SEVERAL MATHEMATICAL METHODS IN WATER POLLUTION CONTROL

Tanehiro Futagami

SEVERAL MATHEMATICAL METHODS IN WATER POLLUTION CONTROL

A THESIS

Presented to

The Faculty of the Graduate Division

By

Tanehiro Futagami

In Partial Fulfillment

of the Requirement for the Degree of

Doctor of Engineering in Sanitary Engineering

Kyoto University

February, 1976

SEVERAL MATHEMATICAL METHODS IN WATER POLLUTION CONTROL

Tanehiro Futagami
Department of Sanitary Engineering
Kyoto University

やまとは
くにの まほろば
たゝなづく 青垣
山ごもれる
やまとし うるはし

To my mother and brother

Acknowledgments

The author wishes to gratefully acknowledge the generous guidance and encouragement of Professor Tomitaro Sueishi at Osaka University, who offered him the opportunity to conduct this investigation. He also wishes to express sincere gratitude to Professor Shigehisa Iwai for cordial guidance and encouragement. To Associate Professor Nobuyuki Tamai at the University of Tokyo he is gratefully indebted for the invaluable guidance, review and encouragement. Many thanks are extended to Professor Yoshiaki Iwasa at Kyoto University and Dr. Mashio Yatsuzuka for cordial guidance and encouragement. He is grateful to Professor Junzaburo Matsunashi and former Associate Professor Susumu Shimizu at Kobe University for cordial guidance and encouragement. He acknowledges the guidance and encouragement received from Dr. Akira Wada at Central Research Institute of Electric Power Industry, and assistance in correcting English provided by Visiting Scientist Harold Solomon at the University of Tokyo. It is a great pleasure to acknowledge the encouragement of KTC (Technology Promotion Club at Kobe University). Grateful acknowledgments are due to numerous people from whom he was received a lot of kindness. Finally, the author wishes to express sincere gratitude for the spiritual backbone to accomplish this work received from late Teacher Zenmon Fujii, the late Chairman of the Publishing Committee on the "History of Toyota-cho" in Yamaguchi prefecture.

The study was performed by utilizing the HITAC 8700/8800 at the computer center of the University of Tokyo and the program library for the simplex method "HI/TC/LP02" (developed by K. Ikura and revised by Hitachi Co. Ltd.).

Abstract

Several fundamental mathematical methods in water pollution control are studied. A finite element & linear programming method (FELP Method, or, the F.E. & L.P. Method), a transient finite element & linear programming method (T. FELP Method) and a finite difference & linear programming method (FDLP Method, or, the F.D. & L.P. Method) have been developed and systematized in order to solve systems of differential equations with both equality or inequality constraints and an objective function. Such systems are frequently encountered in various engineering and scientific problems of control and optimal design and, especially, are of interest in environmental and water resources problems. FELP Method and T. FELP Method have been developed and systematized by the combined use of finite element method and linear programming. In a manner similar to FELP Method, FDLP Method has been systematized. An efficient computational algorithm for the proposed methods is developed by taking note of the fact that the proposed methods have special structures. The applicability of the proposed methods is shown through numerical examples of water pollution problems governed by the convective diffusion equation. The proposed methods make it possible to obtain not only the optimal discharges from the various types of outfall to meet water quality requirements, but also the distribution patterns of several water qualities in the water basin simultaneously. The proposed methods give us new criteria for selecting the locations of outfalls and optimal volumes of discharged waste water. The proposed methods may become useful techniques for analysis, planning and assessment in environmental and water resources problems. An analytical method based on double Fourier series is also developed in order to check the computations of FELP Method and FDLP Method.

TABLE OF CONTENTS

Page

Page

List of Tables

List of Figures

Chapter 1

INTRODUCTION

1-1. Background

The pollution of water in various water basins has presented many obstacles to water resources utilization and danger to ecological balance in our living environment. The water basins have received the assorted and heavey waste of industries and municipalities. The water pollution problems have become very complex and include various factors associated with natural science and social science.

Numerous kinds of water pollution problems have occured in various bodies of water from little streams to large oceans throughout the world. These are thermal, biological, chemical, etc. and are intimately connected with each other. Recently, special attention has been given to the pollution problems in large bodies of water such as lakes and oceans, because these bodies of water have been often used as the destinations of pollutants.

If strong efforts are not made to prevent these serious water pollution problems, they will present more problems in the near future. Integrated efforts to prevent these serious problems are desired.

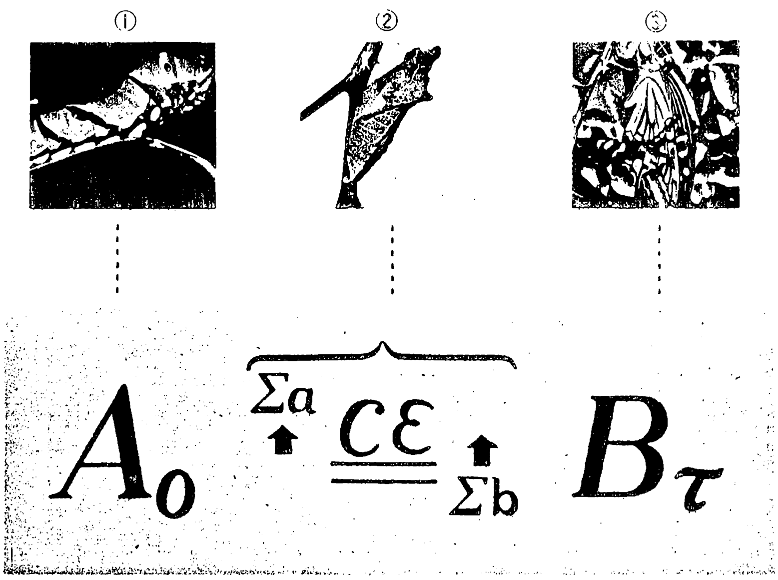
Therefore, research on water pollution phenomena and on water pollution control which offers pertinent data in guiding the prevention of water pollution problems is required. The water pollution problems are so complex that broad approaches are required for analysis and solution. The broad perspective of water pollution problems requires synthetical and novel

approaches as well as ordinary approaches.
Various synthetical and novel approaches such as the systems approach (10), cybernetics (120), equivalent transforming thinking (54), theory of structural stability and morhpogénèse (88), etc. have been developed and applied in various problems.

In this investigation several mathematical methods based on the systems approach are developed and systematized in order to control water pollution problems, especially in large bodies of water.

An Illustration of Equivalent Transforming Thinking developed by K. Ichikawa

創造過程としてみた場合の昆虫の完全変態3段階



from Reference (54)

1-2. Previous Works - Literature Review

The problems of water pollution and water resources include the hydraulics, chemistry and biology of the water basin along with the economic, sociologic and political environments of the water basin. Therefore, a great deal of significant research has been presented by various kind of investigators.

However, the literature review in this section is mainly concerned with the present investigation, that is:

1). *Systems Approach to Problems of Water Pollution and Water Resources*

2). *Numerical Analysis in Continuum Mechanics*

3). *Combined Use of Finite Difference Method with Linear Programming*

4). *Research Associated with Diffusion Phenomena*

Systems Approach to Problems of Water Pollution and Water Resources

Perhaps the most important advance made in recent years in the field of water pollution control and water resources management was the adoptation of systems engineering. Systems engineering, which is sometimes called (incorrectly) operations research, is, in a sence, the attacking of complex problem on a broad front. Variables describing components or states of a systems can be defined, and relationships between them represented, through equations in a mathematical model. These relationships, whether linear or nonlinear, can be properly evaluated by a variety of techniques, some of which have been made possible by the advance in computer science.

However, if the problems were reducible to a set of mathematical expressions, there would have been no need to invent systems engineering: perhaps numerical analysis would have been sufficient in many cases.

But in problems of water pollution control and water resources management there are many factors to be considered. Therefore, systems engineering is frequently adopted in these fields. The systems approach has made remarkable progress in the analysis of components and subsystems from which the synthesis of the complete system may be possible.

In 1955, a water resources program was initiated at Harvard University. The purpose of this program was to develop a methodology for planning and designing complex, modern water resources systems. From the outset, it was recognized that these systems exhibit certain fundamental engineering aspects, as well as broad economic and social overtones. The accepted approach to water resources planning and management was influenced by the publications of the group's work in 1962 (76). Although it was oriented principally toward the traditional "quantity" purposes, it clearly demonstrated the utility of operations research techniques in the design of complex, multi-purpose, multi-constrained water resources systems. In 1963, the Harvard group issued a report which addressed itself specifically to the quality aspects (110).

At about the same time that the Harvard Water Program got under way, and quite independently from it, a number of workers at the University of California began studing the problem of optimization of water resource systems.

Tennessee Valley Authority (S8) has presented numerous significant researches associated with water resources systems through the activities.

The start of the Western Resources Conference in 1959 has accelerated the growth of water resources engineering (67).

The activities of the Committee of Water Resources Systems of the International Association for Hydraulic Research has progressed water resources engineering.

Another landmark in the rapid growth of water research is the start of the Water Pollution Research Conference in 1962.

The accomplishments of the above-mentioned groups and conferences, and Kneese's definitive consideration of water quality economics (66) have provided the stimulas for the most recent research and development in the area of water quality management systems.

Begining in the mid-fifties, several books were published setting out in detail the economic theory relevant water resources development and use and demonstrating its applicability and significance by means of case studies [Krutilla and Eckstein (69), Hirshleifer et al. (53), Kuiper (70) and Hall and Dracup (48)].

Recently, Buras (1972) published a useful book associated with systems approach to water pollution problems (10).

The scientific, professional, and technical literature in this field is rapidly increasing in quantity. It seems that there is an information explosion. The information explosion presents certain problems to the investigator and professional man. Although there is at least one internationally wellknown scientific periodical in this field (*Water Resources Research*), much material appears in a host of other publications.

Though much significant work has been reported in the literature illustrating the mathematical methods in systems engineering, such as linear programming, non-linear programming, dynamic programming, maximum principle, queueing theory (44), distributed parameter theory (S4, S5), and simulation to reservoir and waste treatment systems and hydraulic problems, the literature revived in the reminder of this paragraph will, for the most part, relate directly to the subject of this investigation; that is water pollution control and water resources management.

Linear Programming

Linear programming is concerned with solving a special type of problem: one in which all relations among the variables are linear, both in the constraints and the function to be optimized (13, 16, 39, 84).

Historically, the general problems of linear programming was first developed and applied in 1947 by George B. Dantzig, Marshall Wood, and thier associates of the U.S. Department of the Air Force.

Linear programming has been applied to a great variety of problems of water quality management. Among the first of the water quality systems analysts of the 1960's were Lynn, Thomann, Sobel, Deininger, Sumitomo and Sueishi.

Lynn et al. used linear programming for selecting an optimal combination of unit treatment processes for a waste treatment plant which provides the required degree of treatment at minimum cost (74).

Lynn (1964) also used linear programming in the problems of stage design of treatment plants. The research provides the incremental capacity of a waste treatment plant to be constructed during each of several time periods as well as the financing details (investments schedule, funds borrowed, etc.) (75).

Thomann and Sobel applied linear systems analysis to the problem of water quality management in estuary (109). This was based on mathematical model for dissolved oxygen presented by Thomann in a previous paper (108). Later Thomann (S9) was concerned with obtaining minimum-cost pollution control; and he presented an application to the Delaware Estuary which utilized a model for temporal and spatial variation of dissolved oxygen.

In 1965, Sobel (94) compared a minimum-cost linear programming formulation for regional water quality systems with the traditional uniform treatment approach.

Deininger (14, 15) separated a river basin into reaches bounded by waste discharges and used linear programming to determine treatment levels at each outfall.

In 1965 Sueishi studied prevention of water pollution and sewage systems planning by using linear programming (98). Sueishi and Minamimoto (99) used linear programming in the allocation problems of waste water discharges.

Sumitomo (101), and Goda, Sueishi and Sumitomo (41) used linear programming for the allocation of water volume and water quality in industrial utilization.

Goodman and Dobbins (42) considered the problems of regional cost associated with use of a river for municipal water supply, assimilation of treated waste, and recreation.

Johnson (59) used linear programming to specify waste treatment requirements among dischargers on the Delawarer Estuary.

Loucks and Lynn (73), ReVelle, Loucks and Lynn (89) used linear programming to select waste treatment levels that would satisfy dissolved oxygen constraints at minimum regional cost.

For more recent works the Graves et al. (1972) article (43) and the Arbavi and Elzinga (1975) article (4) are recommended.

Non-Linear Programming

Methods of solving the problems of non-linear objective function with either non-linear or linear constraints is called non-linear programming (44).

Kerri (65) used non-linear programming to determine the minimum-cost solution to maintaining a specified dissolved oxygen concentration in Oregon's Willamette River basin.

Variational Principle

Probably, variational principle is, in a sense, one of the most beautiful expressions for natural phenomena. The variational approach was developed independently by Euler and Lagrange. Variational principle originated from the endeavor to determine extrema or stational values for functionals (12, 28, 117, S1). A functional is defined as a quantity or function which depends upon the entire course or path of one or more functions rather than on a number of discrete variables.

* *

あらゆる自然の行動は、その自然によって、可能なるかぎり短い方法と時間とでなされる。〔D. 4 r.〕

あらゆる自然の行動は最短距離を通っておこなわれる。〔G. 74 v.〕

岩波文庫

5322—5324

レオナルド・ダ・ヴィンチの手記

下

杉浦明平訳

* *

Dynamic Programming and Maximum Principle

Among the numerous attempts to find new dynamic optimization techniques in recent years Bellman's dynamic programming (6) and Pontryagin's maximum principle (87) are probably the two most successful. They are intimately related with variational principle.

Dynamic programming is an effective computational technique to solve the optimal planning problems in the controllable multi-stage decision process, based on Bellman's optimality principle.

Maximum principle deals with optimization of functional subject to differential equation constraints (20, 87). The discrete version of maximum principle has been also studied by several investigators [Fan and Wang (19)].

Hall and Buras (1961) first suggested the use of dynamic programming in water resources problems (47).

Liebman (71) and Liebman and Lynn (72) used discrete dynamic programming to minimize total basin waste treatment costs associated with meeting specified dissolved oxygen stream standards for Willamette River basin.

Meier and Beightler (S6) presented a method for decomposing non-serial river basin systems into equivalent serial systems which can be analyzed by dynamic programming.

Dysart (17) used dynamic programming for water quality planning and management for the case in which there is or could be interaction of pollutants in the stream.

Futagami (35) studied water pollution control by combined use of dynamic programming and classical optimization technique.

Takamatsu, Naito et al. (102) used maximum principle in optimization process of activated sludge.

Numerical Analysis in Continuum Mechanics

Finite element method and finite difference method are the most popular methods in numerical analysis of physical phenomena.

Although recently an international journal in this filed (*International Journal for Numerical Methods in Engineering*) has been published, much material appeaerin a host of other publications. Information explosion in this field is tremendous.

Finite Element Method

Perhaps the most astonishing success in numerical analysis of continuum mechanics (26, 27) in recent years has been finite element method (22, 29, 37, 97, 119, 121, 122).

The finite element method originated in structural mechanics (80, 86, 112). The method adopts matrix-vector forms (32, 78) in the formulation. Among the first of finite element analysts were Zienkiewitcz, Clough, Turner, Martin, Pian and Melosh.

Now other applications of the method are rapidly developing because of its generality.

Finite element method has been extended to fluid mechanics (21, 62, 83). Oden has made energetic research on generalization of finite element method.

Various formulation techniques of finite element method have been developed. Among these techniques Rayleigh-Ritz finite element method based on variational principle and Galerkin finite element method on weighted residual process are the most prevailing ones.

The advantages of finite element method, as compared to other numerical approaches, are numerous. The method is completely general with respect to geometry, material properties and boundary conditions. Arbitrary thermal mechanical, biochemical loads are possible. Mathematically, it can be show that the method converges to the exact solution as the number of elements is increased; therefore any desired degree of accuracy can be obtained. In addition, the matrix equations formulated by the method can be placed in a band form and require much less computer storage and solution time than nonbanded matrix equations. Therefore, finite element method has essentially superseded finite difference method.

Several researches associated with diffusion and conduction problems have been presented (9, 63). Guymon (1970) and Guymon et al. (1970) studied diffusion convection problems by using Rayleigh-Ritz finite element method (45, 46).

Smith, Farraday and O'Connor (1973) used Galerkin finite element method in the study of the diffusion convection phenomena (93).

Finite Difference Method

Until the development of finite element method, finite difference method was the most frequently used method for the solution of partial differential equations. Numerous publications associated with finite difference method have been published (11, 38, 49, 81, 114).

Probably Collatz is the most eminent contributor in this field.

Recently, Naruoka (1974) presented a brief explanation and literature review about finite difference method (81).

Various finite difference algorithm or schemes have been presented for the solution of diffusion convection equation or its similar derivatives [Shamir and Harleman (92), Stone and Brian (96), and Wada (115)].

Combined Use of Finite Difference Method with Linear Programming

Several researches have suggested the requirement of the mixed use of mathematical methods of operations research and solving techniques of differential equations in order to attack more complicated problems of water resources [Bredehoeft and Young (7), Maddock (77), and Martin, Burdak and Young (79)].

In 1974 Aguado and Remson (2) made the combined use of finite difference method with linear programming in the field of ground water management. The article is an excellent pioneering research associated with finite difference & linear programming method. The feasibility of the combined use of finite difference method with linear programming was tested by using simple examples. Cases of confined and unconfined, one and two-dimensional, steady-state and transient flow are presented. The examples presented the feasibility and inherent problems of the approach. In the article objective functions and production constraints were designed to test and display the method most efficiently and not because of hydrologic or economic importance. But the article offerded the possibility of substitution of other objective functions and constraints, either physical or economic, that are relevant to particular field problems.

Research Associated with Diffusion Phenomena

There is an apparent analogy between heat conduction in solids and diffusion phenomena. By taking note of the fact there is an apparent analogy between heat conduction in solids and diffusion phenomena, Fick (1855) first presented the famous mathematical representation for diffusion phenomena, or Fick's equation. Fick's equation was derived by adopting the mathematical equation for heat conduction given by Fourier.

Fourier's great work "Théorie Analytique de la Chaleur" (1822) is bible in this field (25).

Taylor (1921) studied turbulent diffusion from the statistical point of view (106).

Other distinguished contributors in this field were Kármán (60, 61), Richardson (S7), Kolmogoroff and Prandtl.

A great deal of significant research associated with diffusion phenomena has been published such as

in 1950: Alberstone et al. (3)

in 1953: Taylor (107), Batchelor (5)

in 1959: Elder (18)

in 1961: Ippen and Harleman (55), Rohsenow and Choi (90)

in 1962: Abraham (1),

in 1964: Harleman (50), Wiegel (118),

in 1965: Brooks and Koh (8), Iwasa and Imamoto (57)

in 1966: Harleman (51)

in 1967: Fisher (23)

in 1969: Iwai, Inoue and Higuchi (56), Parker and Krenkel (85), Tamai, Wiegel and Tornberg (104)

in 1970: Vigander, Elder and Brooks (113), Hino (S3)

in 1971: Stolzenbach and Harleman (95)

in 1972: Tamai (103)

in 1974: Hayashi (52), Iwasa and Yatsuzuka (58), Kenedy (64), Koh, Brooks, List and Wolanski (68), Neal (82), Tamai (105), Wada (116), Hino (S2)

in 1975: Yatsuzuka (123)

1-3. Purpose and Scope of Present Investigation

The purpose of this investigation is to establish several fundamental mathematical methods for a systems approach in water pollution problems. A finite element & linear programming method (FELP Method, or, the F.E. & L.P. Method), A transient finite element & linear programming method (T. FELP Method), and a finite difference & linear programming method (FDLP Method, or, the F.D. & L.P. Method) are studied in order to control water pollution problems governed by the convective diffusion equation.

FELP Method and T. FELP Method are developed by the combined use of finite element method with linear programming in order to solve systems of differential equations with both equality or inequality constraints and an objective function (30, 33, 34). Such systems are frequently encountered in various engineering and scientific problems of control and optimal design and, especially, are of interest in field problems such as heat conduction, diffusion convection, seepage flow, electric (or magnetic) potential.

Finite element method, originated in structural mechanics, is powerful numerical method for the solution of differential equations, and has been extended to many other physical phenomena because of its generality with respect to geometry, material properties and boundary conditions.

Linear programming is one of the most frequently used mathematical methods of operations research and is widely used in environmental and water resources problems.

In a manner similar to FELP Method, FDLP Method is also systematized by the combined use of finite difference method with linear programming (31).

Finite difference method is one of the most popular methods for the sollution of differential equations and has been used in analysis of various physical phenomena.

Aguado and Remson (2) made a pioneering research associated with FDLP Method in the field of ground water management.

In the development and systematization of proposed methods (FELP Method, T. FELP Method and FDLP Method), the concepts of decision variable and state variable are adopted as in Bellman's dynamic programming or Pontryagin's maximum principle.

Previously, the author studied dynamic programming in a sewage treatment system (36) and the experience at that time has provided many useful suggestions in the development and systematization of the proposed methods in which the concepts of the decision variable and state variable are adopted. In the authors's previous study, besides general computation technique of dynamic programming used other investigators in water pollution control, an efficient computation technique for dynamic programming was presented. The technique was devised by the combined use of dynamic programming with classical optimization technique based on differentiation, and reduced calculation work considerably.

In order to check the computations of FELP Method and FDLP Method, an analytical method to solve the diffusion equation is developed by using double Fourier series. In the analytical method the loads are displaced by double Fourier series so as to satisfy the given boundary conditions. Such a technique based on double Fourier series is often used in structural analysis (24, 40, 111).

In this investigation the applicability of the proposed methods is shown through numerical examples of water pollution problems governed by the convective diffusion equation. The proposed methods make it possible to obtain not only the optimal discharges from the various types of outfall to meet water quality requirements, but also the distribution patterns of several water qualities in the water basin simultaneously. Especially, the proposed methods are studied to control water pollution problems in large bodies of water in which diffusion-convection phenomena predominate.

The five chapters following the Introduction deal with the development, systematization and application of the proposed methods.

It seems that FELP Method based on finite element method is superior to FDLP Method based on finite difference method because of the generality of FELP Method. However, FDLP Method is much easier to understand because of its simplicity in the mathematical formulation. Therefore, at first FDLP Method is described and the description of FELP Method follows it.

In Chapter 2 systematization of FDLP Method and the application to water pollution control are presented.

Chapter 3 is concerned with development and systematization of FELP Method and the application to water pollution control.

In the description of Chapter 2 and 3, efforts are made to ease the comparison between FDLP Method and FELP Method.

In Chapter 4 an efficient computational algorithm for FELP Method and FDLP Method is developed by taking note of the fact that these methods have special structures.

Chapter 5 deals with T. FELP Method which is the extension of FELP Method to the time domain.

In Chapter 6 conclusions of this investigation are presented.

The representative computer programs developed in this investigation are presented in the appendices.

Finally, in the application of the proposed methods, it should be noted that the proposed methods are based on the differential systems. Generally, problems connected with natural phenomena are so complex and sometimes so catastrophic (88) that they are frequently beyond the expression ability of the usual differential systems. And in water pollution problems the same is true. Therefore, the proposed methods can be applied only when and where the differential systems are valid.

* *

There is, however, a point of diminishing returns. The actual world is extremely complicated, and as a matter of fact the more that one studies it the more one is filled with wonder that we have even "order of magnitude" explanations of the complicated phenomena that occur, much less fairly consistent "laws of nature". If we attempt to include too many features of reality in our mathematical model, we find ourselves engulfed by complicated equations containing unknown parameters and unknown functions. The determination of these functions leads to even more complicated equations with even more unknown parameters and functions, and so on. Truly a tale that knows no end.

If, on the other hand, made timid by these prospects, we construct our model in too simple a fashion, we soon find that it does not predict to suit our tastes.

It follows that the Scientist, like Pilgrim, must wend a straight and narrow path between the Pitfalls of Oversimplification and the Morass of Overcomplication.

Richard E. Bellman

* *

References

1. Abraham, G., "Jet Diffusion in Liquid of Greater Density," Journal Hydraulics Division, ASCE, Vol. 86, HY6, 1960, pp. 1-13.
2. Aguado, E. and Remson, I., "Ground-Water Hydraulics in Aquifer Management," Journal of the Hydraulics Division, ASCE, Vol. 100, No. HY1, January, 1974, pp. 103-118.
3. Alberstone, M.L., Dai, Y.B., Jensen, and Rouse, H., "Diffusion of Submerged Jets," Trans., ASCE, Vol. 115, 1950, pp. 639-677.
4. Arbabi, M. and Elzinga, J., "A General Linear Approach to Stream Water Quality Modeling," Water Resources Research, Vol. 11, No. 2, 1975, pp. 191-196.
5. Batchelor, G.K., "The Theory of Homogeneous Turbulence," Cambridge University Press, London, 1953.
6. Bellman, R., "Dynamic Programming," Princeton University Press, 1957.
7. Bredehoeft, D.J. and Young, A.R., "The Temporal Allocation of Ground Water - A Simulation Approach," Water Resources Research, Vol. 6, No. 1, February, 1970, pp. 3-21.
8. Brooks, N.H. and Koh, R.C.Y., "Discharge of Sewage Effluent from a Line Source into a Stratified Ocean," Proceedings, 11th Congress of IAHR., Leningrad, No. 2-19, 1965.
9. Bruch, JR, C.J. and Zyvoloski, G., "Transient Two-Dimensional Heat Conduction Problems Solved by the Finite Element Method," International Journal for Numerical Methods in Engineering, Vol. 8, 1974, pp. 481-494.

10. Buras, N., "The Systems Approach to Water Resources Problems," Scientific Allocation of Water Resources, Water Resources Development and Utilization - A Rational Approach, Chapter 2, American Elsevier Publishing Company, Inc., New York, 1972, pp. 15-29.

11. Collatz, L., "Boundary-Value Problems in Partial Differential Equations," The Numerical Treatment of Differential Equations, Chapter 5, 3rd ed., Springer-Verlag, 1960.

12. Courant, R. und Hilbert, D., "Methoden Mathematishen Physik I, 3. Auflage, Springer-Verlag, 1968, pp. 139-233.

13. Dantzig, B.G., "Linear Programming and Extensions," Princeton University Press, 1963.

14. Deininger, R.A., "The Economics of Regional Pollution Control Systems, Proceedings, 21th Purdue Industrial Waste Conference, 1966, Part II, pp. 815-833.

15. Deininger, R.A., "Water Quality Management: The Planning of Economically Optimal Pollution Control Systems," Thesis Presented to Northwestern University, in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy, June, 1965.

16. Dorfman, R.P., Samuelson, A.P. and Solow, M.R., "Linear Programming and Economic Analysis," McGraw-Hill, New York, 1958.

17. Dysart, C.B., III, "Development and Application of Rational Water Quality Planning Model," Thesis Presented to Georgia Institute of Technology, in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy, Georgia Institute of Technology, January, 1969.

18. Elder, J.W., "The Dispersion of Marked Fluid in Turbulent Shere Flow," Journal of Fluid Mechanics, Vol. 5, May, 1959, pp. 544-560.

19. Fan, L.T. and Wang, C.S., "The Discrete Maximum Principle," Wiley, 1965.

20. Fan, L.T., "The Continuous Maximum Principle," John Wiley & Sons, 1966.

21. Farraday, V.R., O'Connor, A.B. and Smith, I.M., "A Two-Dimensional Finite Element Model for Partially Mixed Estuaries," Proceedings, 16th Congress of IAHR, São Paulo, 1975, C31, pp. 267-274.

22. Finlayson, T., "The Method of Weighted Residuals and Variational Principles," Academic Press, New York, 1972.

23. Fisher, H.B., "The Mechanics of Dispersion in Natural Stream," Journal of the Hydraulics Division, ASCE, Vol. 93, HY6, 1967, pp. 187-216.

24. Flüge, W., "5. 2 Solution of the Inhomogeneous Problem," Stress in Shells, 4th Printing, Springer-Verlag, New York Inc., 1967, pp. 221-226.

25. Fourier, B.J., "Théorie Analytique de la Chaleur, Œubres de Fourier," Darboux, G., ed., Tome Premier, Gauthier-Villars et Fils, Imprimeurs-Libraires, Paris, 1887.

26. Fung, Y.C., "Foundations of Solid Mechanics," Prentice-Hall, 1965.

27. Fung, Y.C., "A First Cource in Continuum Mechanics," Prentice-Hall, 1969.

28. Funk, P., "Variationsrechnung ihre Anwendung in Physik und Technik," Zweite Auflage, Springer-Verlag, 1970.

29. Fujii, H., "Some Remarks on Finite Element Analysis of Time-Dependent Field Problems," Proceedings of 1973 Tokyo Seminar on Finite Element Analysis, University Tokyo Press, November, 1973, pp. 91-106.

30. Futagami, T., "Development of Finite Element & Linear Programming Method and Its Application to Problems of Water Pollution Control," Proceedings, 19th Hydraulics Conference, Japan Society of Civil Engineers, February, 1975, pp. 133-138, (in Japanese).

31. Futagami, T., "Fluid Analysis by Finite Difference & Linear Programming Method," Proceedings, Annual Conference in Kansai Branch, Japan Society of Civil Engineers, April 1975, pp. II-6-1-II-6-2, (in Japanese).

32. Futagami, T., "Matrix," Design Manual of Civil Engineering, Tsuruoka, T., ed., Maruzen Company, Tokyo, 1974, pp. 61-71, (in Japanese).

33. Futagami, T., Tamai, N. and Yatsuzuka, M., "Finite Element & Linear Programming Method in Water Pollution Control," Journal of the Hydraulics Division, ASCE, (in Contribution).

34. Futagami, T., "Finite Element & Linear Programming Method and Water Pollution Control," Proceedings, 16th Congress of the International Association for Hydraulic Research, July-August, 1975, C7, pp. 54-61.

35. Futagami, T., "Research on Prevention of Water Pollution and Sewage System Planning," Thesis Presented to Kobe University, in Partial Fulfillment of the Requirements for Degree of Master of Civil Engineering, Kobe University, February, 1969.

36. Futagami, T., "Dynamic Programming for a Sewage Treatment System," Proceedings, 5th International Water Pollution Research Conference, Jenkins, H.S., ed., Pergamon Press Ltd., Spring 1971, pp. II-21/1-II 21/12.

37. Futagami, T., "Finite Element Method," Design Manual of Civil Engineering, Tsuruoka, T., ed., Maruzen Company, Ltd., Tokyo, 1974, pp. 229-235, (in Japanese).

38. Futagami, T., "Numerical Analysis (Finite Difference and Interpolation)," Design Manual of Civil Engineering, Tsuruoka, T. ed., Maruzen Company, Ltd., Tokyo, 1974, pp. 71-78, (in Japanese).

39. Gass, I.S., "Linear Programming, Method and Applications," 3rd ed., McGraw-Hill Kogakusha, Ltd., 1969.

40. Girkmann, K., "210. Das Beiderseits Frei Auflegende Rohr," XIX. Das Kreiszylindrishe Rohr, Flachentragwerke, Sechste Auflage, Wien-Springer-Verlag, 1963, pp. 473-486.

41. Goda, T., Sueihsi, T. and Sumitomo, H., "Water Volume and Quality in Planning of Industrial Water Utilization, Journal of JSCE, No. 134, 1966, pp. 43-55, (in Japanese).

42. Goodman, A.S. and Dobins, W.E., "Mathematical Model for Water Pollution Control Studies," Journal of the Sanitary Engineering Division, ASCE, Vol. 92, No. SA6, 1966, pp. 1-19.

43. Graves, G., Hatfield, B.G. and Whinston, B.A., "Mathematical Programming for Regional Water Quality Management," Water Resources Research, Vol. 8, 1972, pp. 273-290.

44. Gue, L.R. and Thomas, E.M., "Mathematical Methods in Operations Research," The Macmillan Company, pp. 61-99, pp. 100-158.

45. Guymon, G.L., "A Finite Element Solution of the One-Dimensional Diffusion-Convection Equation," Water Resources Research, Vol. 6, No. 1, 1970, pp. 204-210.

46. Guymon, G.L., Scott,H.V. and Herrman, R.L., "A General Numerical Solution of the Two-Dimensional Diffusion-Convection Equation by the Finite Element Method, Water Resources Research, Vol. 6, No. 6, 1970, pp. 1611-1617.

47. Hall, W.A. and Buras, "The Dynamic Programming Approach to Water Resources Developments," Journal of Geophysical Research 66, 1961, 517.

48. Hall, W.A., and Dracup, J.A., "9. Water Quality Subsystems," Water Resources Systems Engineering, McGraw-Hill, 1970, pp. 341-356.

49. Haraguchi, C., Solution by Finite Difference Method," Rectangular Plates on Elastic Foundation, Sankaido, 1964, pp. 351-366, (in Japanese).

50. Harleman, D.R.F., "The Significance of Longitudinal Dispersion in the Analysis of Pollution in Estuary, Proceedings, 2nd International Water Pollution Research Conference held in Tokyo, August, 1964, Vol. 1, pp. 279-290.

51. Harleman, "12. Diffusion Process in Stratified Flow," Estuary and Coastline Hydrodynamics, Ippen, A.T., ed., McGraw-Hill, 1966, pp. 575-597.

52. Hayashi, T. Miyahara, H. and Arita, M., "A Mathematical Model on the spread of Heated Water Discharged Horizontaly at the Water Surface," Proceedings, U.S.-Japan Joint Seminar on Engineering and Environmental Aspects of Waste Heat Disposal, April, 1974.

53. Hirshleifer, J., Millman, J.W. and D.E. Haven, James, C., "Water Supply: Economics, Technology, and Policy, Cicago, University Cicago Press, 1960.

54. Ichikawa, K., "Science for Creative Thinking - Introduction to Equivalent Transforming Thinking by Illustrations," Japan Broadcasting Publishing Corporation (Nihon Hoso Shuppan Kyokai), 1970, (in Japanese).

55. Ippen, A.T. and Harleman, D.R.F., "One-Dimensional Analysis of Salinity Intrusion in Estuaries, Tech. Bull. No. 5, Comm. on Tidal Hydraulics Corps of Engrs. Vicksburg, Miss., June, 1961.

56. Iwai, S., Inoue, Y. and Higuchi, H., "Survey and Prediction in the Omuta Industrial Harbour, Proceedings, 4th International Water Pollution Research Conference held in Prague, 1969, pp. 883-895.

57. Iwasa, Y. and Imamoto, H., "Turbulent Diffusive Process in Open Channel Flow by Means of Tracer Injection," Proceedings, 11th IAHR Congress, Leningrad, 1965.

58. Iwasa, Y. and Yatsuzuka, M., "Spread of Heated Water from Vertical Multi-Port Diffuser," Proceedings, U.S.-Japan Joint Seminar on Engineering and Environmental Aspects of Heat Disposal, April, 1974.

59. Johnson, E.L., "A Study in the Economic of Water Quality Management," Water Resources Research, Vol. 3, No. 2, 1967, pp. 291-305.

60. Kármán, V.T., "The Fundamentals of Statistical Theory of Turbulence," Journal of the Aeronautical Sciences, Vol. 4, 1937, pp. 131-138.

61. Kármán, V.T., "Turbulence," Twenty-fifth Wilbur Wright Memorial Lecture Delivered to the Royal Aeronautical Society on May 27th, 1937, Aeronautical Reprints, No. 89, 1937.

62. Kawahara, M. et al., "Steady Flow Analysis of Incompressible Viscous Fluid by Finite Element Method," Theory and Practice in Finite Element Structural Analysis, Proceedings of 1973 Tokyo Seminar on Finite Element Analysis, University Tokyo Press, November, 1973, pp. 557-572.

63. Kawai, T., On the Finite Element Analysis of Diffusion Problems," Theory and Practice in Finite Element Structural Analysis, Proceedings of 1973 Tokyo Seminar on Finite Element Analysis, University Tokyo Press, November, 1973, pp. 541-556.

64. Kenedy, F.J., "Plume Recirculation and Interference in Mechanical Draft Cooling Towers," Proceedings, U.S.-Japan Joint Seminar on Engineering and Environmental Aspects of Heat Disposal, April, 1974.

65. Kerri, K.D., "A Dynamic Model for Water Quality Control," Journal of the Water Pollution Control Federation, Vol. 39, No. 5, 1967, pp. 772-786.

66. Kneese, A.V., "The Economics of Regional Water Quality Management," Baltimore, The Johns Hopkins Press, 1964.

67. Kneese, A.V. and Bower, T.B., "Managing Water Quality: Economics, Technology, Institutions," Baltimore, The Johns Hopkins Press, 1968.

68. Koh, R.C.Y., Brooks, N.H., List, E.J. and Wolanski, J.E., "Thermal Outfall Diffusers for the San Onofre Nuclear Power Plant," Report No. KH-R30, W.M. Keck Laboratory, C.I.T., January, 1974.

69. Krutilla, J.V. and Eckstein, O., "Multi Purpose River Development: Studies in Applied Economic Analysis," Baltimore: Johns Hopkins Press, 1958.

70. Kuiper, E., "Water Resources Development," Butterworths, London, 1965.

71. Liebman, J.C., "The Optimal Allocation of Stream Dissolved Oxygen Resources," Ph. D. Dissertation, Cornell University, Ithaca, N.Y., 1965.

72. Liebman, J.C. and Lynn, R.W., "The Optimal Allocation of Stream Dissolved Oxygen," Water Resources Research, Vol. 2, No. 3, 1966, pp. 581-591.

73. Loucks, D.P. and Lynn, R.W., "Probabilistic Models for Predicting Stream Quality," Water Resources Research, Vol. 2, No. 3, 1966, pp. 593-605.

74. Lynn, W.R., Logan, A.J. and Charnes, A., "Systems Analysis for Planning Waste Water Treatment Plants, Journal of the Water Pollution Control Federation, Vol. 34, No. 6, June, 1962, pp. 565-581.

75. Lynn, W.R., "Stage Development of Waste Water Treatment Works," Journal of the Water Pollution Control Federation, Vol. 36, No. 6, June, 1964, pp. 722-751.

76. Maass, A., Hufschmidt, M.M., Dorfman, R., Thomas, A.H., Marglin, S.A. and Fair, M.G., "Design of Water Resources Systems," Harvard University Press, 1962.

77. Maddock, T., III, "Relative Impact of Hydrologic and Economic Factors in Groundwater Management," Thesis Presented to Harvard University, at Cambridge, Mass., in 1971, in Partial Fullfillment of the Requirements for the Degree of Doctor of Philosophy.

78. Martin, C.H., "Introduction to Matrix Methods of Structural Analysis," McGraw-Hill, 1966.

79. Martin, W.E., Burdak, T.G. and Young, R.A., "Projecting Hydrologic and Economic Interrelationships in Groundwater Basin Management," American Journal of Agricultural Economics, Vol. 51, No. 5, December, 1969, pp. 1593-1597.

80. Melosh, R.J., "Structural Analysis of Solids," Proceedings, ASCE, St4, pp. 205-223, August, 1963.

81. Naruoka, M., "5. Finite Difference Method," Essence of Structural Mechanics, Maruzen, Tokyo, 1974, pp. 201-212.

82. Neal, C.L., "Laboratory Studies Spray Cooling Systems," Proceedings, U.S.-Japan Joint Seminar on Engineering and Environmental Aspects of Heat Disposal, April, 1974.

83. Oden. J.T., "A Finite Element Analog of Navier-Stokes Equations," Journal of Engineering Mechanics, Vol. 96, No. EM4, ASCE, 1970, pp. 529-534.

84. Ochard-Hays, W., "Advanced Linear Programming Computing Techniques," McGraw-Hill, 1968.

85. Parker, F. and Krenkel, P.A., "Thermal Pollution Status of the Art," National Center for Research and Training in the Hydrological and Hydraulic Aspects of Water pollution Control, Report No. 3, Department of Environmental and Water Resources Engineering, Vanderbilt University, December, 1969.

86. Pian, T.H.H. and Tong, P., "Basis of Finite Element Methods for Solid Continua," International Journal for Numerical Method in Engineering, Vol. 1, 1969, pp. 3-28.

87. Pontryagin, L.S., Boltyanskii, R.V., Gamkrelidze, R.V. and Mischenko, E.F., "The Mathematical Theory of Optimal Processes," Wiley-Interscience, 1962, (English Translation by Trirogoff)

88. René Thome, "Stabilité Structuelle et Morphogénèse - Essai d'une théorie générale des modèles," W.A. Benjamin, Inc., 1972.

89. ReVelle, C.S., Loucks, P.D. and Lynn, R.W., "A Management Model for Water Quality Control," Journal Water Pollution Control Federation, Vol. 39, No. 7, 1967, pp. 1164-1183.

90. Rohsenow, M.W. and Choi, Y.C., "Heat, Mass and Momentum Transfer," Prentice-Hall, Inc., 1961.

91. Samuelson, A.P., "Introduction," Foundations of Economic Analysis, Chapter 1, Atheneum, New York, 1971, pp. 3-6.

92. Shamir, U.Y. and Harleman, D.R.F., "Numerical Solutions for Dispersion in Porous Mediums," Water Resources Research, Vol. 3, No. 2, 1967, pp. 557-581.

93. Smith, I.M., Farraday, R.V. and O'Connor, "Rayleigh-Ritz and Galerkin Finite Elements for Diffusion-Convective Problems," Water Resources Research, Vol. 9, No. 3, June, 1973, pp. 593-606.

94. Sobel, M.J., "Water Quality Improvement Programming Problems," Water Resources Research, Vol. 1, No. 4, 1966, pp. 477-487.

95. Stolzenbach, K.D. and Harleman D.R.F., "Three-Dimensional Heated Surface Jets," Water Resources Research, Vol. 9, No. 1, 1971, pp. 129-137.

96. Stone, H.L., Brian, T.L.P., "Numerical Solution Convective Transport Problems, Journal American Institute of Chemical Eng., Vol. 9, No. 5, 1963, pp. 681-688.

97. Strang, G. and Fix, J.G., "An Analysis of the Finite Element Method," Prentice-Hall Series in Automatic Computation, Forsythe, G., ed., Prentice-hall, Inc., 1973.

98. Sueishi, T., "Research on Prevention of Water Pollution and Optimization of Sewage Planning," Proceedings, 2nd Symposium on Sanitary Engineering, JSCE, 1965, pp. 86-98, (in Japanese).

99. Sueishi, T. and Minamimoto, S., "Allocation of Pollution Load to Associated Municipalities with a River Basin," Proceedings, 4th Symposium on Sanitary Engineering, Japan Society of Civil Engineers, 1967, pp. 60-67, (in Japanese).

100. Sugiki, A., Matsuo, T. and Tanaka, K., "Water Resources Studies in Ara Valley," Proceedings, 4th IWPR Conference, Prague, 1969, pp. 107-109.

101. Sumitomo, H., "Planning of Industrial Water Utilization by Linear Programming," Proceedings, Annual Conference, JSCE, II-85, pp.85-1 - 85-2 (in Japanese), 1964.

102. Takamatsu, T. and Naito, M. et al., "Research on Optimization of Processes in Activated Sludge (Report No. 2)," Journal of Japan Society of Sewerage and Sewer Work, Vol 5, No. 46, 1968, (in Japanese).

103. Tamai, N., "Unified View of Diffusion and Dispersion in Coastline Waters," Journal of Faculty of Engineering, The University of Tokyo, Vol. XXXI, No. 4, 1972, pp. 531-692.

104. Tamai, N., Wiegel, L.R., Tornberg, F.G., "Horizontal Surface Discharge of Warm Water Jets," Journal of the Power Division, ASCE, Vol. 95, No. PO2, October, 1969, pp. 253-276.

105. Tamai, N., "Dispersion Models in Coastline Waters with Predominant Transverse Shear," Coastal Engineering in Japan, Vol. 17, Japan Society of Civil Engineering, 1974, pp. 185-197.

106. Taylor, G.I., "Diffusion by Continuous Movements," Proceedings, London, Math. Soc., Vol. 20, pp. 196-212, 1921.

107. Taylor, G.I., "Dispersion of Soluble Matter in Solvent Flowing Slowly through a Tube, Proceedings of the Royal Society A, Vol. ccxix, 1953, pp. 186-203, The Sientific Papers of Sir Geoffrey Ingram Taylor, Vol. IV, Mechanics of Fluids: Miscelaneous Papers, Cambridge at the University Press, 1971, pp. 225-243.

108. Thomann, R.V., "Mathematical Model for Dissolved Oxygen," Journal of the Sanitary Engineering Division, American Society of Civil Engineers, Vol. 89, No. SA5, 1963, pp. 1-30.

109. Thomann, R.V. and Sobel, J.M., "Esturine Water Quality Management and Forecasting," Journal of the Sanitary Engineering Division, American Society of Civil Engineers, Vol. 90, No. SA5, 1964, pp. 9-36.

110. Thomas, H.A., Jr., and Burden, "Operations Research in Water Quality Management," Final Report to Public Health Service, 1963.

111. Timoshenko, P.S. and Woinowski-Krieger, S., "28. Navier Solution for Simply Supported Rectangular Plates," Theory of Plates and Shells, 2nd edition, McGraw-Hill, Kogakusha, 1959, pp. 108-113.

112. Turner, M.J., Clough, R.W., Martin, H.C., and Topp, L.J., "Stiffness and Deflection Analysis of Complex Structures," Journal of the Aeronautical Sciences, Vol. 23, No. 9., 1956. pp. 805-823 and p. 854.

113. Vigander, S., Elder, R.A. and Brooks, N.H., "Internal Hydraulics of Thermal Discharge Diffusers," Journal of the Hydraulics Division, ASCE, Vol. 96, HY1, 1970, pp. 509-527.

114. Vitásek, E., "Solution of Partial Differential Equations by the Finite Difference Method," Survey of Applicable Mathematics, Rectorys, K., ed., the M.I.T. Press, 1969, pp. 1109-1124.

115. Wada, A., "A Study on Phenomena of Flow and Thermal Diffusion Caused by Outfall of Cooling Water," Proceedings, 10th Conference Coastal Engineering, Vol. 2, ASCE, Vol. II, ASCE, 1966, pp. 1388-1411.

116. Wada, A., "Study on Prediction Method of Simulation Analysis for Diffusion of Discharged Warm Water," Proceedings, U.S.-Japan Joint Seminar on Engineering and Environmental Aspects of Heat Disposal, April, 1974.

117. Washizu, K., "Variational Methods in Elasticity and Plasticity," Pergamon Press, Oxford, 1968.

118. Wiegel, R.L., "16. Mixing Process," Oceanographical Engineering, Prentice-Hall, 1964, pp. 424-441.

119. Wilson, L.E., "The Determination of Temperatures within Mass Concrete Structures," Report No. 68-17, Structural Engineering Laboratory, University of California, December, 1968.

120. Wiener N., "Cybernetics or Control and Communication in the Animal and Machine," second ed., The M.I.T. Press 1948, Reprinted by University Tokyo Press, 1972.

121. Zienkiewicz, O.C., "The Finite Element Method in Engineering Science," 2nd ed., McGraw-Hill, 1971.

122. Zienkiewicz, O.C. and Parekh, C.J., "Transient Field Problems: Two-Dimensional and Three-Dimensional Analysis by Isoparametric Finite Elements," International Journal for Numerical Methods in Engineering, Vol. 2, 1970, pp. 61-71.

123. Yatsuzuka, M., "Hydraulic Research on Spread of Heated Warm Water from Vertical Multi-Port Diffuser," Thesis Presented to Kyoto University, in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy, February, 1975, (in Japanese).

Supplementary References

S1. Forray, J. Marvin, "Variational Calculus in Science and Engineering," McGraw-Hill Book Company, 1968.

S2. Hino, M., "A Discussion on Sampling Problem of Diffusing Plumes," Proceedings, U.S.-Japan Seminar on Engineering and Environmental Aspects of Waste Heat Disposal, Printed at the Department of Civil Engineering, Kyoto University, 1974.

S3. Hino, M., "An Essay on Turbulent Diffusion Due to Random Waves," Proceedings, 14th Japanese Conference on Hydraulics, JSCE, pp. 25-29, (in Japanese).

S4. Hullett, W., "Optimal Aeration: An Application of Distributed Parameter Control Theory," Applied Mathematics and Optimization Vol. 1, No. 1, Springer-Verlag, New York Inc., 1974, pp. 20-63.

S5. Lions, L.J., "Côntrole Optimal de systèmes gouvernés par des équations aux dérivees partielles, Gauthier-Villars, 1968.

S6. Meier, W.L., Jr., and Beightler C.S., "An Optimization Method for Branching Multistage Water Resources," Water Resources Research, Vol. 3, No. 3, 1967, pp. 645-652.

S7. Richardson, L.F., "Atmospheric Diffusion Shown a Distance-Neighbour Graph," Proceedings, Royal Society A, Vol. 110, pp. 709-737.

S8. Tenessee Valley Authority, Division of Water Control Planning Engineering Laboratory,""TVA"s Program for Monitoring Water Temperatures in The Vicinity of Stream Power Plants," Proceedings, U.S.-Japan Seminar on Engineering and Environmental Aspects of Waste Heat Disposal, Printed at the Department of Civil Engineering, Kyoto University, 1974.

S9. Thomann, R.V., "Recent Results from a Mathematical Model of Water Pollution Control in The Delaware Estuary," Water Resources Research, Vol. 1, No. 3, 1963, pp. 349-359.

Chapter 2

FINITE DIFFERENCE & LINEAR PROGRAMMING METHOD IN WATER POLLUTION CONTROL

Summary

Water pollution control is studied by a finite difference & linear programming method (FDLP Method, or, the F.D. & L.P. Method). FDLP Method has been developed by the combined use of a finite difference method with linear programming in order to solve systems of differential equations with both equality or inequality constraints and an objective function. The applicability of FDLP Method is shown through numerical examples of water pollution problems governed by the convective diffusion equation. The method makes it possible to obtain not only the optimal discharges from the various types of outfall to meet water quality requirements, but also the distribution patterns of several water qualities in the water basin simultaneously. A new criterion for selecting the locations of outfalls and the optimal volumes of discharged waste water may be given by the method. The method may become an useful technique for analysis, planning and assessment in environmental and water resources problems. In order to check the results of FDLP Method, an analytical method by double Fourier series is developed and described.

2-1. General Concepts

Water pollution control is studied by a finite difference & linear programming method (FDLP Method, or, the F.D. & L.P. Method). FDLP Method has been developed by the combined use of a finite difference method with linear programming in order to solve systems of differential equations with both equality or inequality constraints and an objective function. Such systems are frequently encountered in various engineering and scientific problems of control and optimal design and, especially, are of interest in environmental and water resources problems.

Aguado and Remson made the combined use of finite difference method with linear programming in the field of ground water management (1).

The finite difference method (3, 11, 23) is one of the most popular methods for the solution of differential equations and has been used in analyses of various physiscal phenomena. Moreover, linear programmin (4) is one of the most frequently used mathematical methods of operations research and is widely used in environmental and water resources problems (5, 18, 21). In the development of FDLP Method the concepts of the decision variable and state variable are adopted as in Bellman's dynamic programming (2, 6, 10) or Pontryagin's maximum principle (7, 17). FDLP Method utilizes the advantages of established numerical techniques of both finite difference method and linear programming. The problems of FDLP Method with many variables and constraints could be solved by many existing computer programs for linear programming.

The applicability of FDLP Method is shown through numerical examples of water pollution problems governed by the convective diffusion equation. A great deal of significant research on water pollution problems associated with diffusion phenomena has been presented (13, 14, 15, 16, 19, 20, 24) and has provided the stimulas and many useful ideas for the application of FDLP Method to water pollution control. FDLP Method makes it possible to obtain not only the optimal discharges (the decision variable) from the various types of outfall to meet with water quality requirements, but also the distribution patterns of water quality (the state variable) in the water basin simultaneously. The method may become one of the useful techniques for the analysis, planning and assessment in environmental and water resources problems.

In order to check the computations of FDLP Method, an analytical method based on double Fourier series is developed and also described.

2-2. Finite Difference & Linear Programming Method

2-2-1. Systems of Basic Differential Equations

FDLP Method has been developed to solve the following systems of differential equations with both equality or inequality constraints and an objective function. (See Fig. 2-1).

Equilibrium Equations

Governing Equation (in the whole domain Ω_s)

$$D.E.\ \left(x_k,\ \phi,\ \frac{\partial\phi}{\partial x_k},\ \ldots,\ \frac{\partial^n\phi}{\partial x^n},\ \theta\right) = 0 \tag{1}$$

Boundary Conditions (on the boundaries S)

$$h\ (x_k,\ \phi) = 0 \tag{2}$$

Constraints (in the subdomains Ω_s^g)

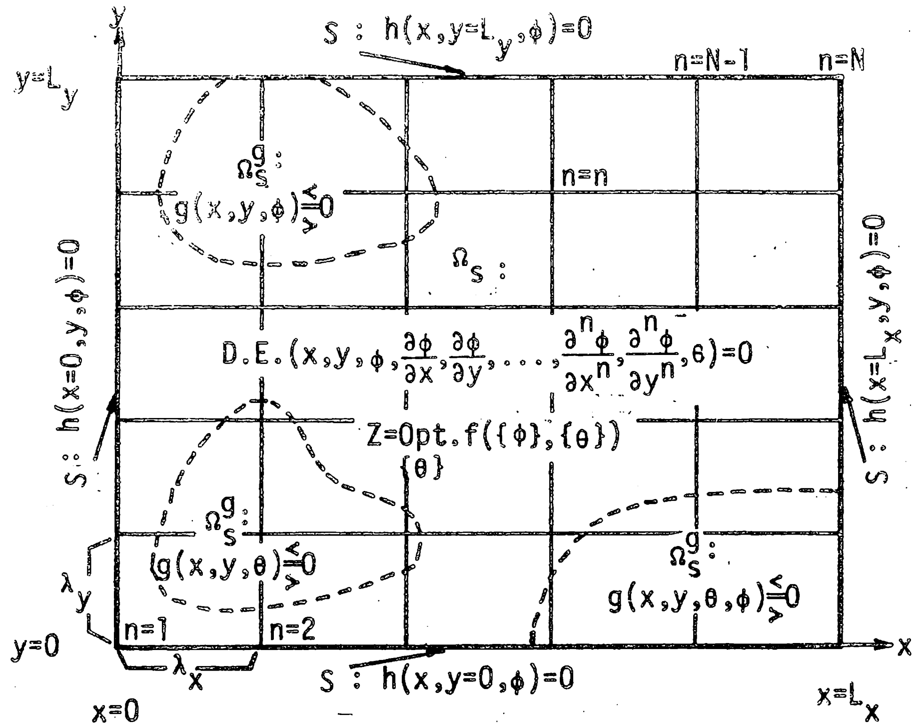
$$g\ (x_k,\ \phi,\ \theta) \lesseqqgtr 0 \tag{3}$$

Objective Function (throughout the whole domain Ω_s)

$$Z = \underset{\{\theta\}}{Opt.}\ f\ (\{\phi\},\ \{\theta\}) = \begin{cases} \underset{\{\theta\}}{Max.}\ f\ (\{\phi\},\ \{\theta\}) \\ \underset{\{\theta\}}{Min.}\ f\ (\{\phi\},\ \{\theta\}) \end{cases} \tag{4}$$

in which $\phi = \phi(x_k)$ = the state variable; $\theta = \theta(x_k)$ = the decision variable; and x_k = Cartesian coordinate (x, y or z).

In the systems governed by two-dimensional second-order differential equations for example, Eq. 1 is expressed as follows:



D.E. : Governing Equation

h : Boundary Condition

g : Constraints

Z : Objective Function

Ω_s : Whole Domain (Whole Water Basin)

S : Boundary

Ω_s^g : Subdomain Associated with Constraints

Fig. 2-1. General Concepts of Finite Difference & Linear Programming Method

$$\underbrace{c_1 \frac{\partial^2\phi}{\partial x^2} + c_2 \frac{\partial^2\phi}{\partial x \partial y} + c_3 \frac{\partial^2\phi}{\partial y^2} + c_4 \frac{\partial\phi}{\partial x} + c_5 \frac{\partial\phi}{\partial y} + c_6\phi}_{\phi\text{-terms}} + \underbrace{c_7\theta}_{\theta\text{-term}} + \underbrace{b}_{const} = 0 \qquad (1)$$

The examples of the boundary conditions are as follows:

$$\begin{aligned} &\phi(x=0,\ y) = C_1(y), \qquad \phi(x=L_x,\ y) = C_2(y), \\ &\frac{\partial\phi}{\partial y}\Big|(x,\ y=0) = 0, \qquad \frac{\partial\phi}{\partial y}\Big|(x,\ y=L_y) = 0 \end{aligned} \qquad (2)$$

As for the constraints, the following simple inequalities are frequently encountered.

$$\begin{aligned} &\underline{\Phi} \leq \phi < \overline{\Phi}, \ or \ \begin{cases} \phi \geq \underline{\Phi} \\ \phi \leq \overline{\Phi} \end{cases} \\ &\underline{\Theta} \leq \theta \leq \overline{\Theta}, \ or \ \begin{cases} \theta \geq \underline{\Theta} \\ \theta \leq \overline{\Theta} \end{cases} \end{aligned} \qquad (3)$$

in which $\underline{\Phi}$ = the lower limit of the state variable; $\overline{\Phi}$ = the upper limit of the state variable; $\underline{\Theta}$ = the lower limit of the decision variable; $\overline{\Theta}$ = the upper limit of the decision variable.

2-2-2. Formulation of FDLP Method

The finite difference method is used in order to discretize the above-mentioned systems as systems of linear algebraic equations. (As for the details, see the next section). Then, the following matrix-vector forms of FDLP Method are obtained and the application of linear programming is possible.

Equilibrium Equations (N-Eqs.)

$$\underset{(N\times N)}{[\,A\,]}\{\phi_n\} + \underset{(N\times I)}{[\,D\,]}\{{}_j\theta_i\} = \{b_n\} \tag{5}$$

Constraints (L_T-Eqs.)

$$\underset{(L_T\times N)}{[\,G_\phi\,]}\{\phi_n\} + \underset{(L_T\times I)}{[\,G_\theta\,]}\{{}_j\theta_i\} \;\substack{\leq\\=\\>}\; \{b^g_l\} \tag{6}$$

Nonnegative Conditions

$$\phi_n \geqq 0 \;\; (n = 1 \sim N), \qquad {}_j\theta_i \geqq 0 \;\; (i = 1 \sim I) \tag{7}$$

Objective Function

$$Z = \underset{\{{}_j\theta_i\}}{Opt.}\; f\,(\{\phi_n\}, \{{}_j\theta_i\}) = \underset{\{{}_j\theta_i\}}{Opt.}\left(\sum_{n=1}^{N} c^\phi_n\,\phi_n + \sum_{i=1}^{I} c^\theta_i\,{}_j\theta_i\right) \tag{8}$$

in which [A] = the state matrix, derived from finite difference method and corrected according to the boundary conditions; [D] = the decision matrix, sparse matrix; $[G_\phi]$ = the state-constraint matrix, generally sparse matrix; $[G_\theta]$ = the decision-constraint matrix, generally sparse matrix; ${}_j\theta_i$ = ith decision variable; j = mesh point number associated with ith decision variable; ϕ_n = nth state variable (state variable at the mesh point n); b_n = constant in nth equilibrium equation; b^g_l =

constant in lth constraint; c_n^{ϕ} = state-evaluation coefficient associated with ϕ_n; c_i^{θ} = decision-evaluation coefficient associated with ${}_j\theta_i$; $n = 1 \sim N$ (N: total number of the state variables, i.e., total number of the mesh points); $i = 1 \sim I$ (I: total number of the decision variables); and $l = 1 \sim L_T$ (L_T: total number of the constraints).

Therefore, FDLP Method is one that optimizes the objective function under the conditions of the equilibrium equations and the constraints.

Since all of the variables in linear programming have to be nonnegative because of the limitation in the computational algorithm based on the simplex method (4), the conditions of Eq. 7 are required.

In FDLP Method the number of the variables is $(N+I)$, the number of the equilibrium equations is N, and the number of the constraints is L_T, respectively. In the sense of general linear programming, the equilibrium equations of FDLP Method are also the constraints. Thus, FDLP Method is a kind of linear programming in which the number of the variables is $(N+I)$ and the number of the constraints is $(N+L_T)$, respectively.

In FDLP Method the solution of the decision variables and the solution of the state variables are obtained simultaneously by the simplex method.

2-3. Water Pollution Control by FDLP Method

2-3-1. Systems of Basic Equations in Diffusion-Convection Phenomena

The basic equation systems of diffusion-convection phenomena with constraints and an objective function are as follows:

Equilibrium Equations

Governing Equation (in the whole water basin Ω_s)

$$\underbrace{\sum_{k=1}^{2 \text{ or } 3} \left(\frac{\partial}{\partial x_k} D_{xk} \frac{\partial \phi}{\partial x_k} - v_k \frac{\partial \phi}{\partial x_k}\right) - K\phi}_{\phi\text{-terms}} + \underbrace{Q^c}_{\theta\text{-term}} + \underbrace{Q^u}_{const} = 0 \tag{9}$$

Boundary Conditions (on the boundaries S)

$$\phi = C_0 \tag{10}$$

$$\frac{\partial \phi}{\partial n} = 0 \tag{11}$$

Constraints (in the subdomains of the water basin Ω_s^g)

$$\phi \leqq \overline{\Phi} \tag{12}$$

$$Q^c \leqq \overline{Q^c} \tag{13}$$

Nonnegative Conditions

$$\phi \geqq 0, \quad Q^c \geqq 0 \tag{14}$$

Objective Function (throughout the whole water basin Ω_s)

$$Z = \underset{\{Q^c\}}{Opt.}\ f\,(\{\phi\},\ \{Q^c\}) \simeq \underset{\{Q^c\}}{Max.}\ \Sigma\, Q^c \tag{15}$$

in which ϕ = the state varaible, i.e., water quality (e.g., temperature or concentration in the water basin); $Q^c = \theta$ = the decision variable, i.e., controllable load (heated discharge from multi-port diffusers (25) or pollutant issued from waste outfalls); $\overline{\Phi}$ = the upper limit of the state variable (water quality requirement); $\overline{Q^c}$ = the upper limit of the controllable load arising from the conditions in the outfalls; D_{xk} = diffusion coefficient (D_x, D_y or D_z); v_k = convective velocity (v_x, v_y or v_z); K = heat transfer coefficient at the water surface or decay factor of pollutant; and Q^u = constant = uncontrollable load (inevitably or naturally generated source or sink, existing unexcludable discharge).

Although, generally the objective function may be composed of the controllable loads $\{Q^c\}$ and water quality distribution $\{\phi\}$ in the water basin, the maximization of the total of the controllable loads is sought in the numerical examples for simplicity. From the view point of the assimilation capacity of the environment, such an objective function gives us the upper limit of the total acceptable loads in the water basin.

As for the constraints, although only the upper limits of the state variable and the decision variable (controllable load) are imposed in the above systems, the following lower limit of the decision variable may become necessary with respect to the conditions of the problems. Such a condition may occasionally arise from the hydraulic conditions in the outfalls or from the treatment efficiencies in plants.

$$Q^c \geq \underline{Q^c} \tag{16}$$

2-3-2. Formulation of FDLP Method in Water Pollution Control

The application of FDLP Method to the systems of the diffusion-convection phenomena mentioned above yields the following matrix-vector forms.

Equilibrium Equations (N-Eqs.)

$$\underset{(N\times N)}{[\ A\ ]}\{\phi_n\} + \underset{(N\times I)}{[\ D\ ]}\{{}_jQ_i^c\} = -\{Q_n^u\} \tag{17}$$

Constraints ((L_T = L+I)-Eqs.)

$$\underset{(L\times N)}{[\ g^{\phi}\ ]}\{\phi_n\} \leqq \{{}_m\overline{\Phi}_l\} \tag{18}$$

$$\underset{(I\times I)}{[\ g^{\theta}\ ]}\{{}_jQ_i^c\} \leqq \{{}_j\overline{Q}_i^c\} \tag{19}$$

Nonnegative Conditions

$$\phi_n \geqq 0\ (n = 1 \sim N),\qquad {}_jQ_i^c \geqq 0\ (i = 1 \sim I) \tag{20}$$

Objective Function

$$Z = \underset{\{{}_jQ_i^c\}}{Opt.}\ f\,(\{\phi_n\},\ \{{}_jQ_i^c\}) = \underset{\{{}_jQ_i^c\}}{Opt.}\ \left(\sum_{n=1}^{N} c_n^{\phi}\,\phi_n + \sum_{i=1}^{I} c_i^{\theta}\,{}_jQ_i^c\right)$$

$$\simeq \underset{\{{}_jQ_i^c\}}{Max.}\ \sum_{i=1}^{I} {}_ja_i\ {}_jQ_i^c \tag{21}$$

in which [A] = the state matrix; [D] = the decision matrix; $[g^{\phi}]$ = the sub-state-constraint matrix; $[g^{\theta}]$ = the sub-decision-constraint matrix, unit matrix; ${}_jQ_i^c = {}_j\theta_i$ = ith controllable load (ith decision variable); j = mesh point number fitted for the location of ith controllable load;

ϕ_n = water quality at the mesh point n (nth state variable); Q^u_n = uncontrollable load at the mesh point n; ${}_j a_i = c^\theta_i$ = area governed by mesh point j; ${}_j\overline{Q}^c_i$ = upper limit of ith controllable load; ${}_m\overline{\Phi}_l$ = lth water quality requirement; m = regulated mesh point number in lth water quality requirement; $i = 1 \sim I$ (I:total number of the controllable loads, i.e., total number of mesh points fitted for the locations of the outfalls of the controllable loads); $n = 1 \sim N$ (N: total number of the state variables, total number of the mesh points); $l = 1 \sim L$ (L: total number of the regulated mesh points in water quality requirements); and L_T = $L + I$ = total number of the constraints.

In the consideration of the loads, it should be noted that in FDLP Method, as in finite difference method, distributed loads are fundamental and concentrated loads are displaced by the equivalent distributed loads. Therefore, in FDLP Method all of the loads expressed in ${}_jQ^c_i$ and Q^u_n are distributed loads. The distributing area of these loads are ${}_ja_i$ (area governed by the mesh point j) and a_n (area governed by the mesh point n), respectively.

In the maximization problem of total of the controllable loads, all of the state-evaluation coefficients $\{c^\phi_n\}$ are equal to zero and all of the decision-evaluation coefficients $\{c^\theta_i\}$ are equal to ${}_ja_i$ as shown in Eq. 21.

Eqs. 18 and 19 are equal to the following equations.

$$\phi_m \leq {}_m\overline{\Phi}_l \quad (l = 1 \sim L) \tag{18'}$$

$${}_jQ^c_i \leq {}_j\overline{Q}^c_i \quad (i = 1 \sim I) \tag{19'}$$

As for the details of the matrices [A], [D], $[g^\phi]$ and $[g^\theta]$, see the next section and Eqs. 34-36.

2-3-3. Discretization by Finite Difference Method

The finite difference method is one of the most popular method for the numerical solution of partial differential equations and the basic concept of the method is simple (3, 11, 23).

The solution domain is subdivided by a net with a finite number of the mesh points N as shown in Fig. 2-1 and the derivatives at each mesh point are replaced by finite difference approximations. In fact the function is sought in the neighbourhood of the given typical point 0 by an interpolating polynomial and derivatives are computed from this porinomial representation. (See Fig. 2-2(a)). Thus, the first derivative and second derivative of these polynomials at the typical point 0 are replaced by the following formulae.

$$\frac{\partial\phi}{\partial x} = \frac{\phi_1 - \phi_2}{2\lambda_x} + O(\lambda_x^2), \qquad \frac{\partial\phi}{\partial y} = \frac{\phi_3 - \phi_4}{2\lambda_y} + O(\lambda_y^2) \tag{22}$$

$$\frac{\partial^2\phi}{\partial x^2} = \frac{\phi_1 - 2\phi_0 + \phi_2}{\lambda_x^2} + O(\lambda_x^2), \qquad \frac{\partial^2\phi}{\partial y^2} = \frac{\phi_3 - 2\phi_0 + \phi_4}{\lambda_y^2} + O(\lambda_y^2) \tag{23}$$

Substituting Eqs. 22 and 23 into Eq. 9 and neglecting the residual terms $O(\lambda_x^2)$ and $O(\lambda_y^2)$, Eq. 9 is rewritten in the following linear algebraic equation.

$$D_x \frac{\phi_1-2\phi_0+\phi_2}{\lambda_x^2} + D_y \frac{\phi_3-2\phi_0+\phi_4}{\lambda_y^2} - v_x \frac{\phi_1-\phi_2}{\lambda_x} - v_y \frac{\phi_3-\phi_4}{\lambda_y} - K\phi_0 + {}_0Q_i^c + Q_0^u = 0 \tag{24}$$

Arranging the above equation, we obtain the following equation at the typical point 0.

$$a_{00}\phi_0 + a_{01}\phi_1 + a_{02}\phi_2 + a_{03}\phi_3 + a_{04}\phi_4 + {}_0Q_i^c = -Q_0^u \tag{25}$$

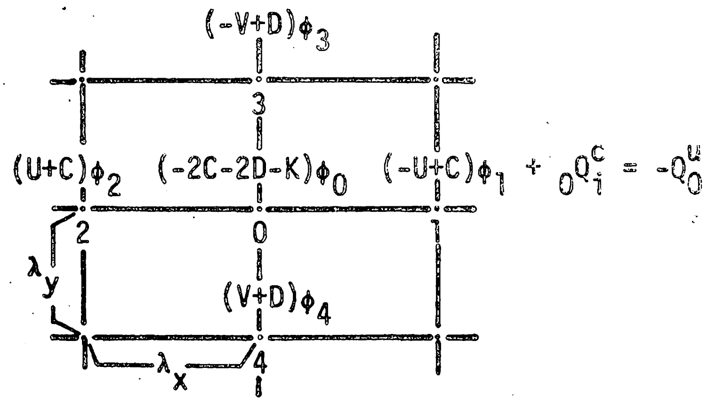


Fig. 2-2(a). Equilibrium Equation at a Typical Point

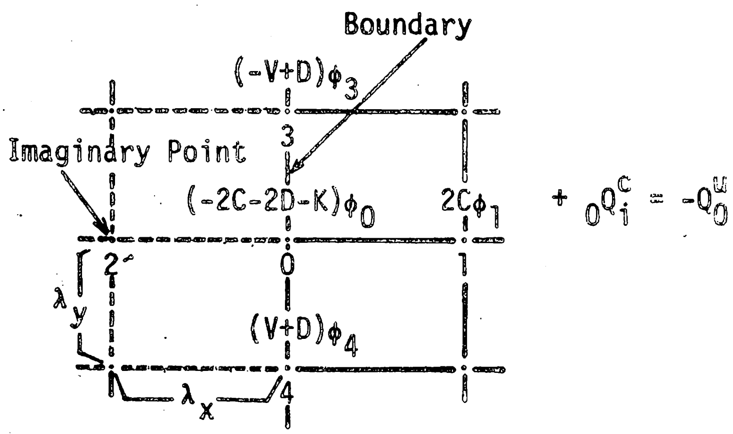


Fig. 2-2(b). Equilibrium Equation on a Non-Conductive Boundary ($\partial\phi/\partial x = 0$ and $v_x = 0$)

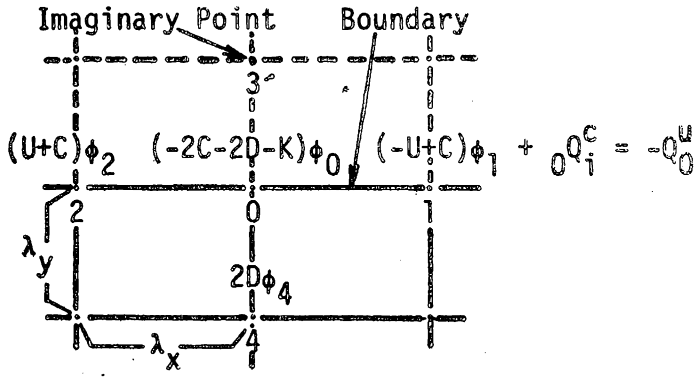


Fig. 2-2(c). Equilibrium Equation on a Non-Conductive Boundary ($\partial\phi/\partial y = 0$ and $v_y = 0$)

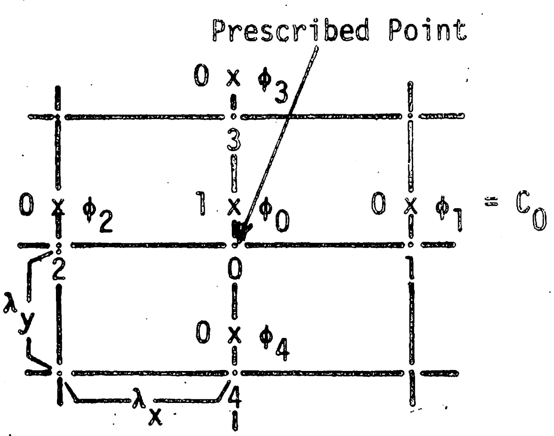


Fig. 2-2(d). Equilibrium Equation at the Prescribed Point ($\phi_0 = C_0$)

, or,

$$(-2C-2D-K)\phi_0 + (-U+C)\phi_1 + (U+C)\phi_2 + (-V+D)\phi_3 + (V+D)\phi_4 + {}_0Q_i^c = -Q_0^u \quad (26)$$

in which

$$U = v_x/2\lambda_x, \quad V = v_y/2\lambda_y, \quad C = D_x/\lambda_x^2, \quad D = D_y/\lambda_y^2 \quad (27)$$

If the derivatives are replaced by finite difference expressions as indicated, one obtains a systems of N algebraic equations for determining the approximate values of the unknown function in N different points of the net.

However, it is necessary to correct the systems of the algebraic equations so as to satisfy the given boundary conditions. Consideration on the boundary conditions yields the following corrections in the coefficients of the equilibrium equations.

On the non-conductive boundary where $\partial\phi/\partial x = 0$ *and* $v_x = 0$

The following correction is made by considering an imaginary point $2'$. (See Fig. 2-2(b)).

$$a_{00}\phi_0 + 2a_{01}\phi_1 + a_{03}\phi_3 + a_{04}\phi_4 + {}_0Q_i^c = -Q_0^u \quad (28)$$

, or,

$$(-2C-2D-K)\phi_0 + 2C\phi_1 + (-V+D)\phi_3 + (V+D)\phi_4 + {}_0Q_i^c = -Q_0^u \quad (29)$$

On the non-conductive boundary where $\partial\phi/\partial y = 0$ *and* $v_y = 0$

The following correction is made by considering an imaginary point $3'$. (See Fig. 2-2(c)).

$$a_{00}\phi_0 + a_{01}\phi_1 + a_{02}\phi_2 + 2a_{04}\phi_4 + {}_0Q_i^c = -Q_0^u \quad (30)$$

, or,

$$(-2C-2D-K)\phi_0 + (-U+C)\phi_1 + (U+C)\phi_2 + 2D\phi_4 + {}_0Q_i^c = -Q_0^u \tag{31}$$

At the point with the prescribed value of $\phi_0 = C_0$ (See. Fig. 2-2(d)).

The correction to the matrix is to make one whole row zero, with the exception of a '*1*' in a_{00}

$$a_{00}\phi_0 = C_0 \tag{32}$$

, or,

$$1 \times \phi_0 = C_0 \tag{33}$$

In order to reduce the number of the decision variables from N to I, ${}_0Q_i^c$ in Eqs. 24, 25, 26, 28, 29, 30 and 31 should be dropped, as shown in Eq. 17, at the mesh point where the outfall for the controllable load does not exist. The decision matrix [D] = $[d_{ni}]$ in Eq. 17 is composed of zero elements with the exceptions of '*1*' in I elements whose row number is j and whose column number is i (see. Eq. 34).

Thus the discretized equilibrium equations (Eq. 17 in the previous section) are obtained.

2-4. Explanation by a Simple Model

In order to clarify the features of FDLP Method, the systems are written down for a simple square model basin in a specific form with non-conductive boundaries, or $\partial\phi/\partial n = 0$ on the four surrounding boundaries as shown in Fig. 2-3. The obtained matrix-vector forms of FDLP Method are as follows:

$$
\begin{array}{c}
\\ 1\\2\\3\\4\\5\\6\\7\\8\\9\\1\\2\\3\\4\\5\\6\\7\\8
\end{array}
\begin{array}{c}
\begin{array}{cccccccccccc} 1&2&3&4&5&6&7&8&9&1&2&3 \end{array}\\
\left[\begin{array}{ccccccccc|ccc}
 & & & & & & & & & & & \\
 & & & & & & & & & 1 & & \\
 & & & & & & & & & & & \\
 & & & & A & & & & & & & \\
 & & & & (9\times 9) & & & & & & 1 & \\
 & & & & & & & & & & D & \\
 & & & & & & & & & & (9\times 3) & \\
 & & & & & & & & & & & 1 \\
 & & & & & & & & & & & \\
\hline
 & & & 1 & & & & & & & & \\
 & & & & & 1 & & & & & \text{Zero} & \\
 & & & & g^{\phi} & & 1 & & & & \text{Matrix} & \\
 & & & & (5\times 9) & & & 1 & & & (5\times 3) & \\
 & & & & G_{\phi} & & & & 1 & & G_{\theta} & \\
\hdashline
 & & & & (8\times 9) & & & & & 1 & (8\times 3) & \\
 & & & & \text{Zero Matrix} & & & & & & \overset{1}{g^{\theta}} & \\
 & & & & (3\times 9) & & & & & & (3\times 3) & 1
\end{array}\right]
\end{array}
\left\{\begin{array}{c}
\phi_1\\ \phi_2\\ \phi_3\\ \phi_4\\ \phi_5\\ \phi_6\\ \phi_7\\ \phi_8\\ \phi_9\\ {}_2Q^c_1\\ {}_5Q^c_2\\ {}_8Q^c_3
\end{array}\right\}
$$

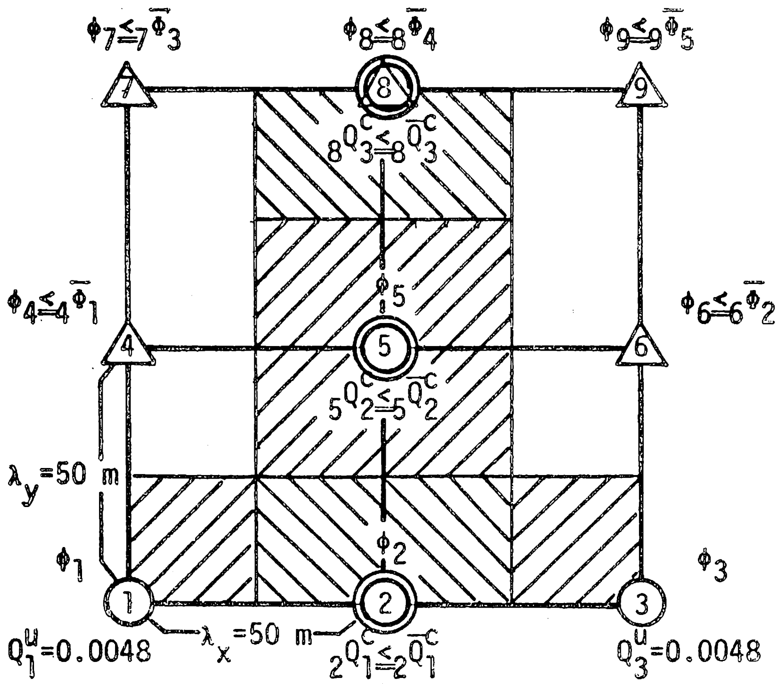
$$
\begin{array}{lll}
= -Q^u_1 & (= -0.0048) & \\
= -Q^u_2 & (= 0) & \\
= -Q^u_3 & (= -0.0048) & \\
= -Q^u_4 & (= 0) & \\
= -Q^u_5 & (= 0) & (34)\\
= -Q^u_6 & (= 0) & \\
= -Q^u_7 & (= 0) & \\
= -Q^u_8 & (= 0) & \\
= -Q^u_9 & (= 0) & \\
\leqq {}_4\overline{\Phi}_1 & (= 3.0) & \\
\leqq {}_6\overline{\Phi}_2 & (= 3.0) & \\
\leqq {}_7\overline{\Phi}_3 & (= 3.0) & (35)\\
\leqq {}_8\overline{\Phi}_4 & (= 3.0) & \\
\leqq {}_9\overline{\Phi}_5 & (= 3.0) & \\
\leqq {}_2\overline{Q}^c_1 & (= 0.0240) & \\
\leqq {}_5\overline{Q}^c_2 & (= 0.0120) & (36)\\
\leqq {}_8\overline{Q}^c_3 & (= 0.0240) &
\end{array}
$$

$$\phi_n \geqq 0 \ (n = 1 \sim 9), \qquad {}_jQ^c_i \geqq 0 \ (i = 1 \sim 3) \qquad (37)$$

$$Z = Max. \sum_{i=1}^{I} {}_j a_i \, {}_j Q_i^c = Max. \, (1250.0 \, {}_2 Q_1^c + 2500.0 \, {}_5 Q_2^c + 1250.0 \, {}_8 Q_3^c) \tag{38}$$

The units of the water quality and the loads are not described, because the method is applicable to general physical problems irrespective of the variables considered.

Computed results for the controllable loads $\{{}_j Q_i^c\}$, water qualities $\{\phi_n\}$ and objective function Z are shown in Fig. 2-4. It should be noted that the distribution pattern of the water quality in Fig. 2-4 arises from the resultant loads composed of the obtained controllable loads $\{{}_j Q_i^c\}$ and the given uncontrollable loads $\{Q_n^u\}$.



$\partial\phi/\partial n = 0$ (On the Four Boundaries)

◎: Mesh Point for Controllable Load

○: Mesh Point for Uncontrollable Load

△: Regulated Mesh Point in Water Quality Requirement (${}_m\overline{\Phi}_l = 3.0,\ l = 1 \sim 5$)

$D_x = 1.0\ m^2/sec$

$D_y = 1.0\ m^2/sec$

$K = 0.001\ {}^1/sec$

$V_x = V_y = 0$

$I = 3 \quad L = 5$

$N = 9$

$L_T = L + I$
$= 5 + 3 = 8$

${}_ja_i \times {}_j\overline{Q}_i^c = 30.0$
$(i = 1 \sim 3)$

${}_2\overline{Q}_1^c = {}_8\overline{Q}_3^c = 0.0240$

${}_5\overline{Q}_2^c = 0.0120$

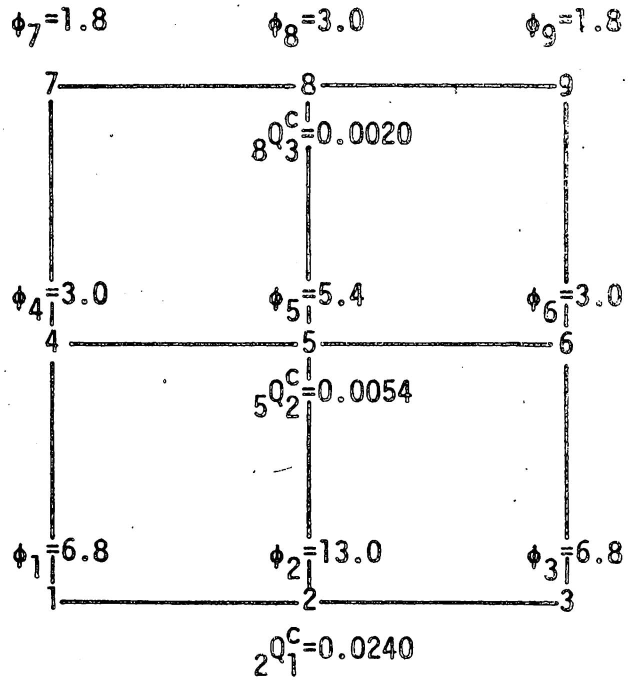
$a_n \times Q_n^u = 3.0$
$(n = 1, 3)$

${}_2a_1 = {}_8a_3 = 1250\ m^2$

${}_5a_2 = 2500\ m^2$

$a_1 = a_3 = 625\ m^2$

Fig. 2-3. Input Data on FDLP Method in a Simple Model Basin



$$
\begin{aligned}
Z &= {}_2a_1\ {}_2Q_1^c + {}_5a_2\ {}_5Q_2^c + {}_8a_3\ {}_8Q_3^c \\
&= 1250 \times 0.0240 + 2500 \times 0.0054 + 1250 \times 0.0020 \\
&= 30.0 + 13.4 + 2.5 = 45.9
\end{aligned}
$$

Fig. 2-4. Results of FDLP Method in a Simple Model Basin

2-5. Numerical Examples in a Model Basin

Numerical examples of FDLP Method are conducted in a rectangular model basin as shown in Fig. 2-5. Two runs are done under the different conditions as shown in Table 2-1. However, the input data shown in Fig. 2-5 and described below are the same in both runs.

Consider the case in which there will be further increases of steady waste discharges over the existing steady waste discharges.

The existing steady waste discharges, or the uncontrollable loads, are, Q^u_{43}, Q^u_{60}, Q^u_{77} and Q^u_{94}, at the mesh points *43*, *60*, *77* and *94*, respectively. They are known loads and their magnitude is equally *0.000480*, or, $a_n \times Q^u_n = 62500.0 \times 0.000480 = 30.0$ ($n = 43, 60, 77$ and 94).

Further increase of the outfalls for the waste discharges due to the increased power of the plants is planned. The locations of the outfalls are planned at the six mesh points, or, *42*, *59*, *76*, *93*, *110* and *127*. The waste discharges issued from these six outfalls are the controllable loads to be solved, or, ${}_{42}Q^c_1$, ${}_{59}Q^c_2$, ${}_{76}Q^c_3$, ${}_{93}Q^c_4$, ${}_{110}Q^c_5$ and ${}_{127}Q^c_6$. The imposed upper limit of the controllable loads ${}_jQ^c_i$ is equally *0.000480*, or, ${}_ja_i \times {}_jQ^c_i = 62500.0 \times 0.000480 = 30.0$ ($i = 1 \sim 6$). Twenty eight mesh points are regulated in the same water quality requirement, or, ${}_m\overline{\Phi}_l = 3.0$ ($l = 1 \sim 28$).

We would like to know which part of the increased waste discharges should be allocated to these six outfalls to meet the water quality requirement and also to know the maximum of the acceptable volume of the increased waste discharges. The maximum should be known from the stand point of environmental assessment and is given by the total of the controllable loads, or the objective function.

The solutions of the controllable loads and objective function in both runs are shown in Table 2-2. The sets of the solution of controllable loads $\{{}_jQ_i^c\}$ give us the optimal locations for outfalls shown by the mesh point number j and the optimal volumes shown by the $\{{}_jQ_i^c\}$ themselves.

The distribution patterns of the water quality obtained by the resultant loads composed of the controllable loads $\{{}_jQ_i^c\}$ and the uncontrollable loads $\{Q_n^u\}$ are also shown in Figs. 2-6(a) and 2-6(b).

The units of the water quality and the loads are not attached in order to show that the method is applicable to various water qualities.

Table 2-1. Input Data on Current Speed and Boundary Conditions

Conditions (1)		Run 1 (2)	Run 2 (3)
Current Speed		$v_x = 0,\quad v_y = 0$	$v_x = 0.03$ m/sec, $v_y = 0$
Boundary Conditions	Boundary 1	$\partial\phi/\partial y = 0$	$\partial\phi/\partial y = 0$
	Boundary 2	$\partial\phi/\partial x = 0$	$\phi = 0.1$
	Boundary 3	$\partial\phi/\partial x = 0$	$\phi = 1.5$
	Boundary 4	$\partial\phi/\partial y = 0$	$\partial\phi/\partial y = 0$

Table 2-2. Controllable Loads and Objective Function

Controllable Load Number i (1)	Associated Mesh Point j (2)	Run 1 Controllable Load: Obtained Distributed Load ${}_jQ_i^c$ (3)	Run 1 Controllable Load: Equivalent Concentrated Load ${}_ja_i^* \times {}_jQ_i^c$ (4)	Run 2 Controllable Load: Obtained Distributed Load ${}_jQ_i^c$ (5)	Run 2 Controllable Load: Equivalent Concentrated Load ${}_ja_i^* \times {}_jQ_i^c$ (6)
1	42	0.000000	0.0	0.000480	30.0
2	59	0.000000	0.0	0.000135	8.4
3	76	0.000022	1.4	0.000000	0.0
4	93	0.000480	30.0	0.000480	30.0
5	110	0.000480	30.0	0.000480	30.0
6	127	0.000480	30.0	0.000480	30.0
Total (Objective Function)			$Z = 91.4$		$Z = 128.4$

*Area Governed by Mesh Point j, ${}_ja_i = 250 \times 250 = 62500$ m^2 ($i = 1 \sim 6$)

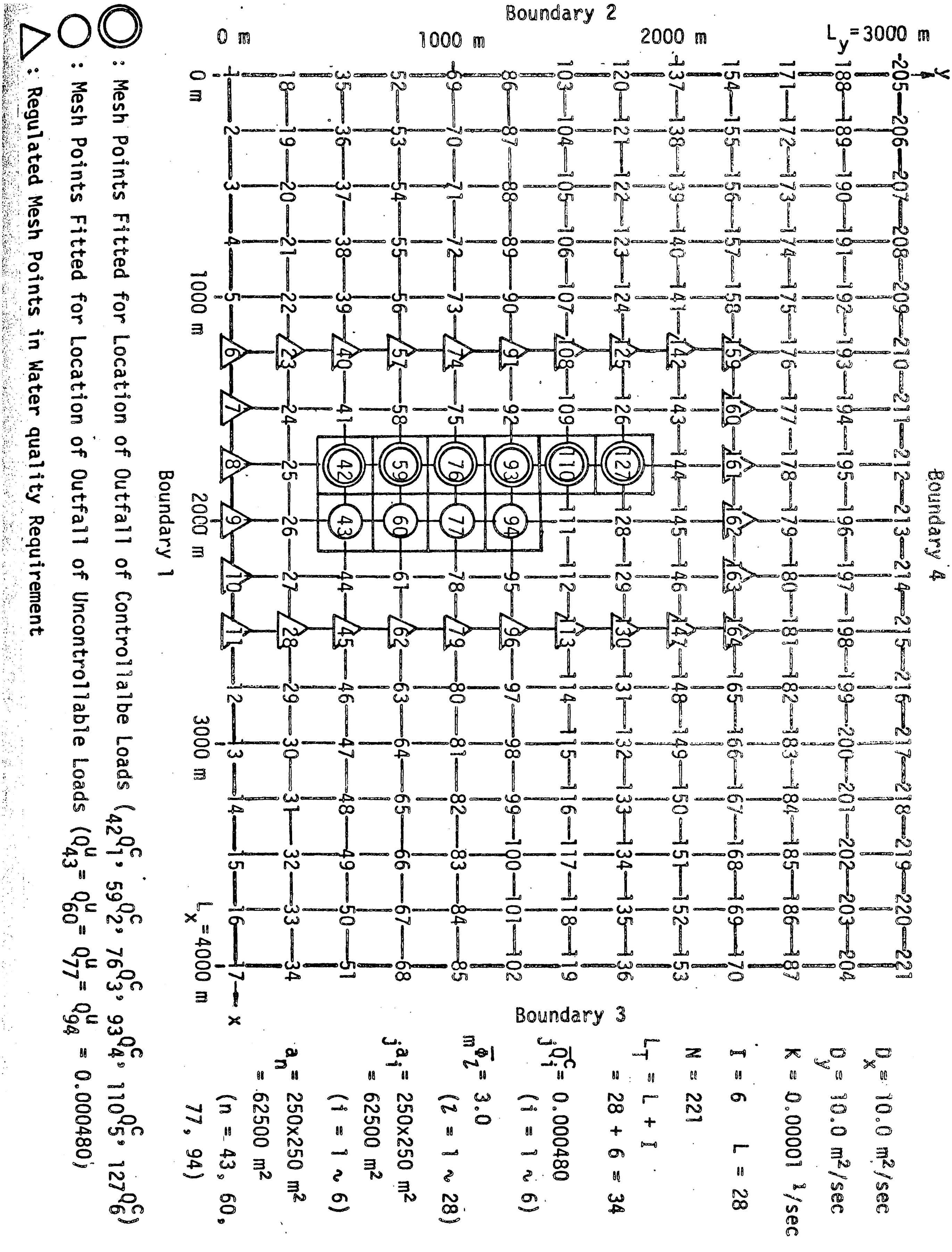


Fig. 2-5. Input Data on FDLP Method in a Rectangular Model Basin

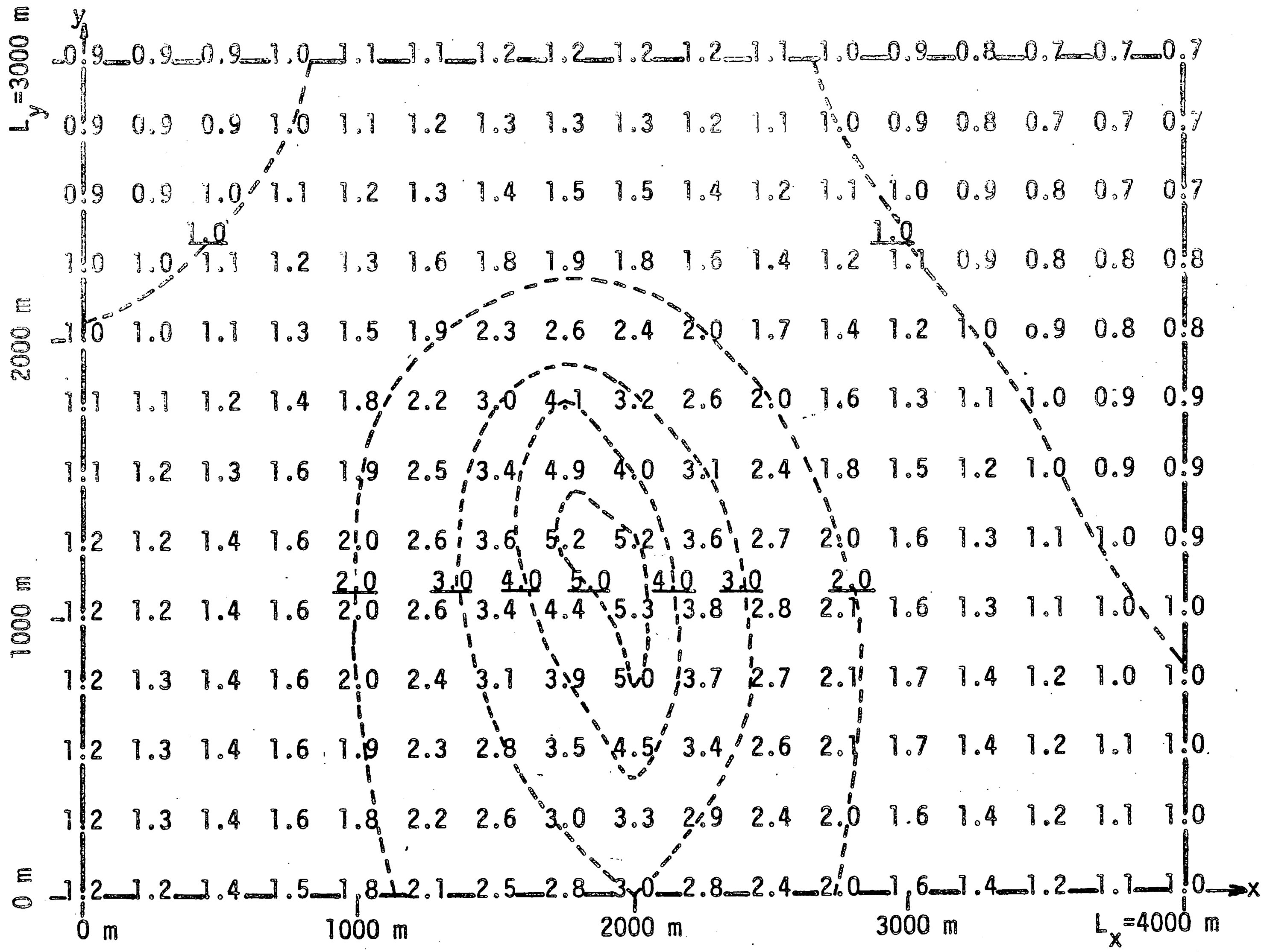


Fig. 2-6(a). Results of FDLP Method in Run 1 (Distribution Pattern of Water Quality)

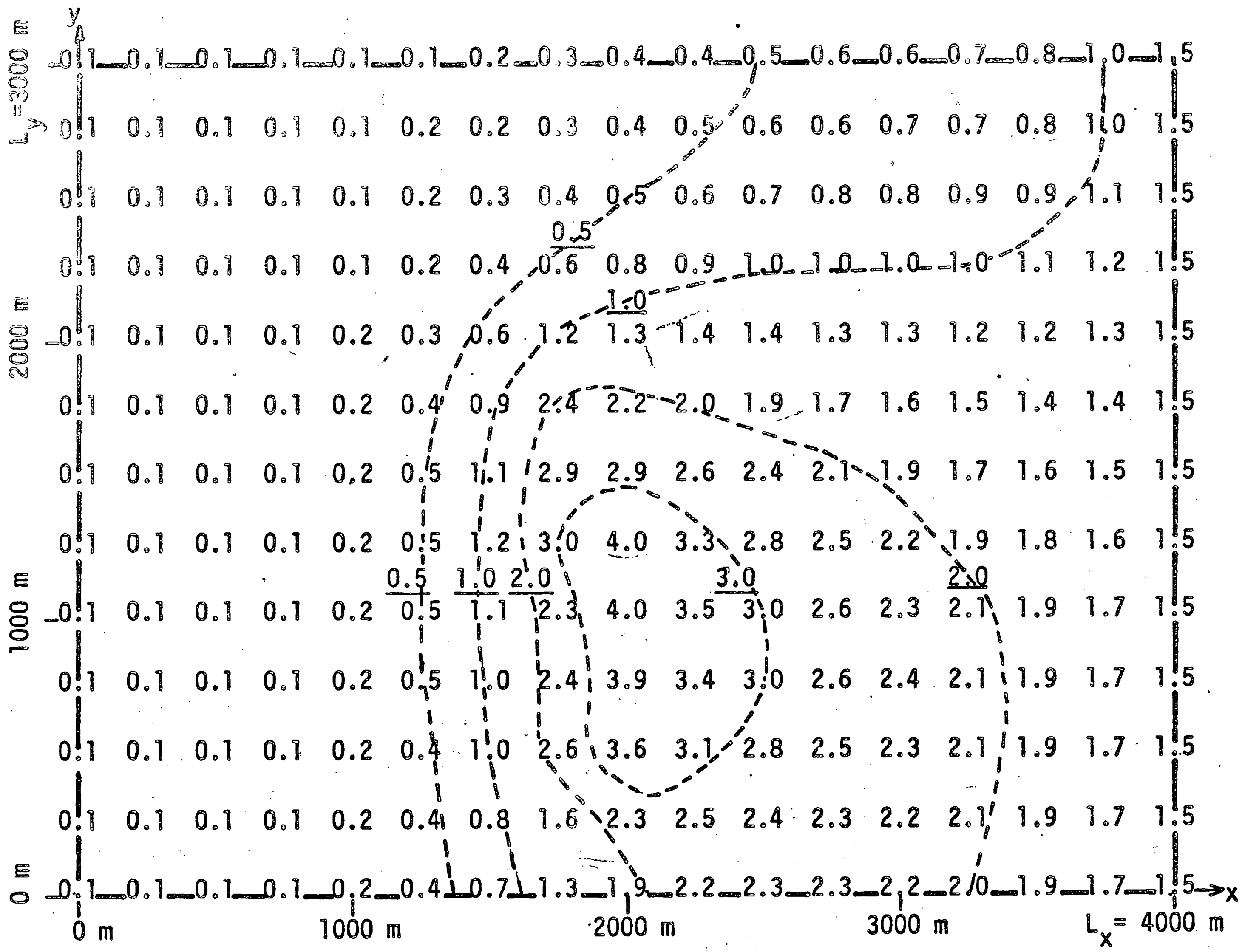


Fig. 2-6(b). Results of FDLP Method in Run 2 (Distribution Pattern of Water Quality)

2-6. Comparison with Analytical Method

2-6-1. Analytical Method by Double Fourier Series

An analytical method by Fourier series (9) is considered in order to check the computations of FDLP Method.

Consider the following diffusion equation with non-conductive boundaries.

$$D_x \frac{\partial^2 \phi}{\partial x^2} + D_y \frac{\partial^2 \phi}{\partial y^2} - K \phi + Q(x,y) = 0 \tag{39}$$

It is desirable to have a solution of Eq. 39 for a given load $Q(x,y)$. Such a solution can be found if the load is distributed according to the following formula:

$$Q(x,y) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} Q_{mn} \cos \frac{m\pi}{L_x} x \cos \frac{n\pi}{L_y} y \tag{40}$$

where Q_{mn} is a constant and expressed in the following equation.

$$Q_{mn} = \frac{2}{L_x} \int_0^{L_x} \frac{2}{L_y} \int_0^{L_y} Q(x,y) \cos \frac{m\pi}{L_x} x \cos \frac{n\pi}{L_y} y \, dy \, dx \tag{41}$$

$$(m = 0,\ 1,\ 2,\ \ldots,\ \infty);\ (n = 0,\ 1,\ 2,\ \ldots,\ \infty)$$

Such a technique in which the load is displaced by double Fourier series so as to satisfy the given boundary conditions is used in structural analyses (8, 22). The first solution of the problem of bending of simply supported rectangular plates and the use for this purpose of double Fourier series are due to Navier, who presented a paper on this subject to French Academy in 1820 (22).

Introducing Eq. 40 into Eq. 39, it may be seen that there exists a solution in the following form

$$\phi = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \phi_{mn} \cos \frac{m\pi}{L_x}x \cos \frac{n\pi}{L_y}y \qquad (42)$$

with unknown constant ϕ_{mn}.

Substituting Eqs. 40 and 42 into Eq. 39 and dropping the trigonometric factors, the following expression for ϕ_{mn} is obtained.

$$\phi_{mn} = \frac{Q_{mn}}{D_x \left(\frac{m\pi}{L_x}\right)^2 + D_y \left(\frac{n\pi}{L_y}\right)^2 + K} \qquad (43)$$

Substitution of the above expression into Eq. 42 yields the following solution.

$$\phi = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \frac{Q_{mn}}{D_x \left(\frac{m\pi}{L_x}\right)^2 + D_y \left(\frac{n\pi}{L_y}\right)^2 + K} \cos \frac{m\pi}{L_x}x \cos \frac{n\pi}{L_y}y \qquad (44)$$

2-6-2. Check Analysis of Run 1 by Analytical Method

The distribution pattern of the water quality in Run 1 obtained by FDLP Method in the previous section as shown in Fig. 2-6(a) is compared with that obtained by the analytical method mentioned above.

The input data used with the analytical method are shown in Fig. 2-7. The input loads $Q(x,y)$ in the analytical method are the resultant loads composed of the given uncontrollable loads $\{Q_n^u\}$ and the obtained controllable loads $\{{}_jQ_i^c\}$ in Run 1. The maximum number for both m and n adopted in the check analysis is *50* and double precision is used.

The results obtained by the analytical method are shown in Fig. 2-8 and show good agreement with those obtained by FDLP Method.

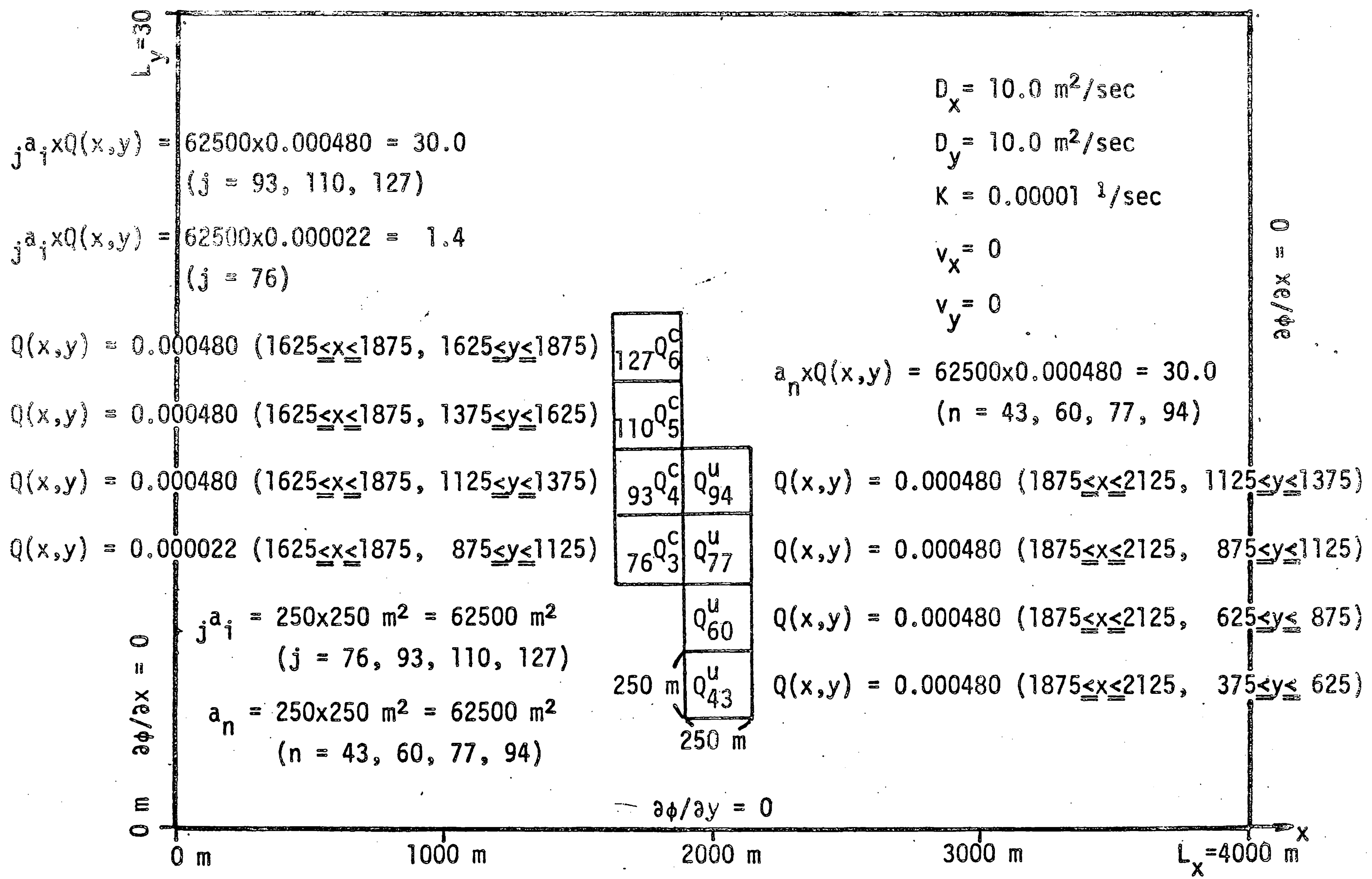


Fig. 2-7. Input Data for Check of Run 1 by Analytical Method

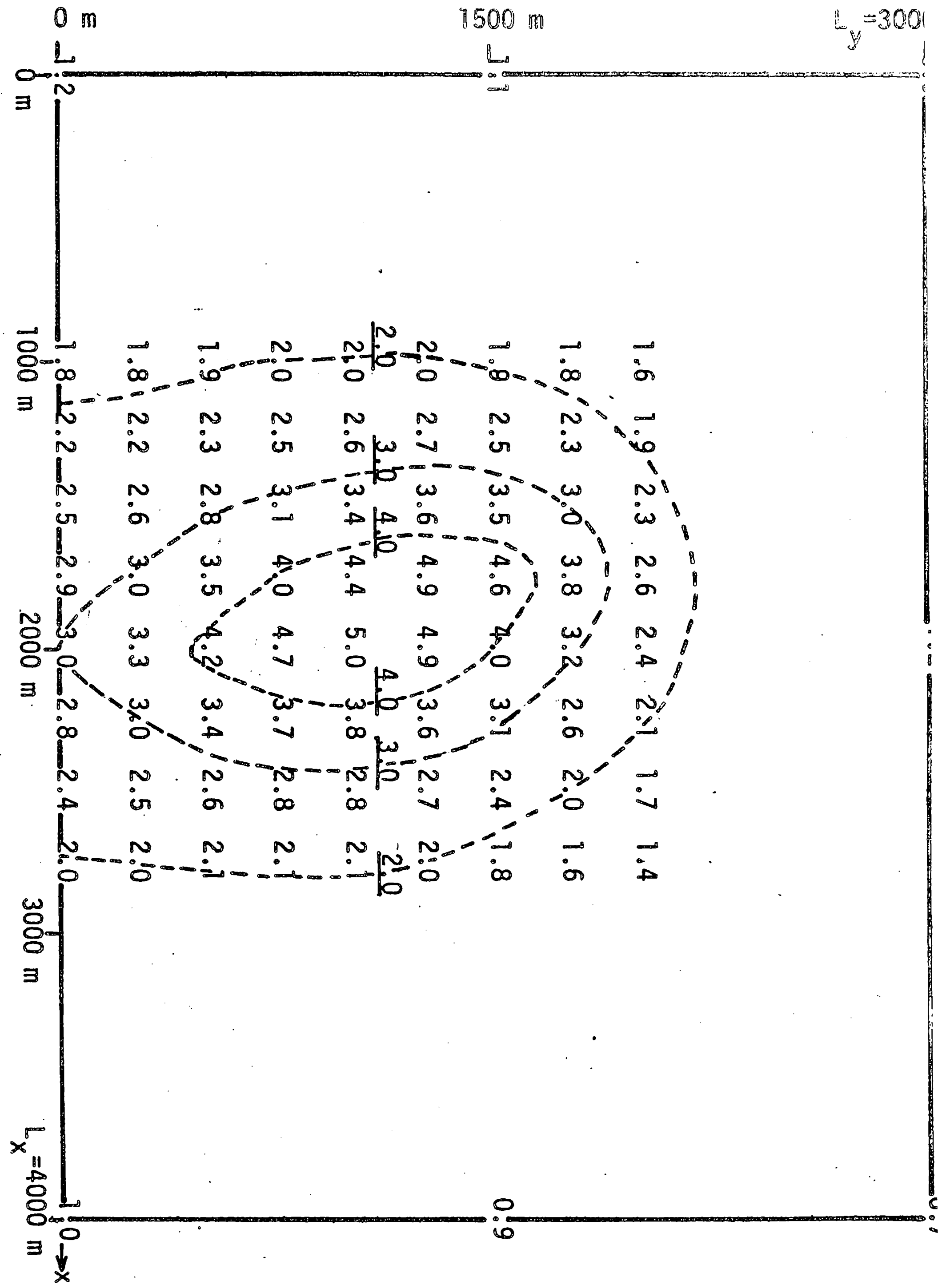


Fig. 2-8. Results of Check of Run 1 by Analytical Method (Distribution Pattern of Water Quality)

2-7 Concluding Remarks

Finite differnce & linear programming method (FDLP Method, or, the F.D. & L.P Method) in water pollution control was described. Some numerical examples in model basins were also presented. The computations of FDLP Method were checked by an analytical method based on double Fourier series.

A new criterion for selecting the locations of outfalls and the optimal volumes of discharged waste water may be given by FDLP Method.

The tractability in both the boundary conditions and the equality or inequality constraints makes sure that FDLP Method becomes one of the most useful techniques for several new types of boundary value problems.

Most practical applications of linear programming make use of the degital computer and existing computer codes. However, in order to save computer time and memory, an efficient computational algorithm of FDLP Method has been developed by taking note of the fact that the method has special structures. The details are presented in Chapter 4.

In a manner similar to FDLP Method, a finite element & linear programming method (FELP Method, or, the F.E. & L.P Method) (12) has also been developd by the combined use of a finite element method with linear programming in the discretization of the systems of differential equations to systems of linear algebraic equations. (See Chapter 3).

Finally, the problems to be attacked from now on in the applications of FDLP Method should be mentioned. Extension of the method to the time domain is necessary to solve transient problems. (See. Chapter 5). The related methods such as finite difference & non-linear programming method and finite difference & integer programming method could be developed.

The developments probably make it possible to solve more complicated problems. Comparison with other analytical methods, experiments and field data should be extended to make the applicability of the method wider.

The computational work in this chapter was performed by using the computer center of the University of Tokyo and its program library for the simplex method "HI/TC/LP02 (made by K. Ikura and revised by Hitachi Co. Ltd.)".

References

1. Aguado, E. and Remson, I, "Ground-Water Hydraulics in Aquifer Management," Journal of the Hydraulics Division, ASCE, Vol. 100, No. HY1, January, 1974, pp. 103-118.
2. Bellman, R., "Dynamic Programming," Princeton University Press, 1957, p. 81.
3. Collatz, L., "The Numerical Treatment of Differential Equations, 3rd ed., Springer-Verlag, 1960.
4. Dantzig, B.G., "Linear Programming and Extensions," Princeton University Press, 1963, pp. 94-119.
5. Deininger, R.A., "Water Quality Management-The Planning of Economically Optimal Pollution Control Systems, Thesis Presented to Northwestern University, in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy, June, 1965.
6. Dysart, C.B., III and Hines, W.W., "Control of Water Quality in a Complex Natural System," Paper Presented at the Institute of Electical and Electronics Engineers Systems Science and Cybernetics Conference, Philadelphia, October 22-24, 1969.
7. Fan, L.T., "The Continuous Maximum Principle," John Wiley & Sons, 1966.
8. Flüge, W., "5.2 Solution of the Inhomogeneous Problem," Stress in Shells, 4th Printing, Splinger Verlag New York Inc., 1967, pp. 221-226.
9. Fourier, "Propagation de la Chaleur dans un Solide Rectangulaire Infini," Théorie Analytique de la Chaleur, Œubres de Fourier, Darboux, G., ed., Tome Premier, Gauthier-Villars et Fils, Imprimeurs-Libraires, Paris, 1887, pp. 141-238.

10. Futagami, T., "Dynamic Programming for a Sewage Treatment System," Proceedings, 5th International Water Pollution Research Conference, Jenkins, H.S., ed., Pergamon Press Ltd., spring 1971, pp. II-21/1-II-21/12.

11. Futagami, T., "Numerical Analysis (Finite Difference and Interpolation)," Design Manual of Civil Engineering, Tsuruoka, T., ed., Maruzen Company, Ltd., Tokyo, 1974, pp. 71-78, (in Japanese).

12. Futagami, T., "Finite Element & Linear Programming Method and Water Pollution Control," Proceedings, 16th Congress of the International Association for Hydraulic Research, July-August, 1975, C7, pp. 54-61.

13. Hayashi, T. and Shuto, N., "Diffusion of Warm Water Jets Discharged Horizontally at the Water Surface," Proceedings, 12th Congress of International Association for Hydraulic Research, Vol. 4, June, 1967, pp. 47-59.

14. Harleman, D.R.F., "Innovations in Heat Disposal in the Oceans," 2nd Annual Sea Grant Lectures and Symposium (October, 1973), MIT, Sea Grant Program Report, MIT, SG. 74,7, 19 pages.

15. Iwasa, Y and Yatsuzuka, M., "Spread of Heated Water from Vertical Multi-Port Diffuser," Proceedings, U.S.-Japan Joint Seminar on Engineering and Environmental Aspects of Heat Disposal, April, 1974.

16. Koh, R.C.Y., Brooks, N.H., List, E.J. and Wolanski, J.E., "Thermal Outfall Diffusers for the San Onofre Nuclear Power Plant, "Report No. KH-R30, W.M. Keck Laboratory of Hydraulics and Water Resources, C.I.T., January, 1974.

17. Pontryagin, L.S., Boltyanskii, R.V., Gamkrelidze, R.V. and Mischenko, E.F., "The Mathematical Theory of Optimal Processes, Wiley Interscience, 1962, (English Translation by Trirogoff).

18. Sueishi, T. and Minamimoto, S., "Allocation of Pollutant Load to Associated Municipalities with a River Basin," Proceedings, 4th Symposium on Sanitary Engineering, JSCE, 1967, pp. 60-67, (in Japanese).

19. Tamai, N., "Unified View of Diffusion and Dispersion in Coastline Waters," Journal of the Faculty of Engineering, the University of Tokyo, Vol. XXXI, NO. 4, 1972, pp. 531-692.

20. Tamai, N., "Dispersion Models in Coastline Waters with Predominant Transverse Shear," Coastal Engineering in Japan, Vol. 17, Japan Society of Civil Engineers, 1974, pp. 185-197.

21. Thomann, R.V. and Sobel, M.J., "Esturine Water Quality Management and Forecasting," Journal of the Sanitary Engineering Division, ASCE, Vol. 90, No. SA5, 1964, pp. 9-36.

22. Timoshenko, P.S. and Woinowsky-Krieger, S., "28. Navier Solution for Simply Supported Rectangular Plates," Theory of Plates and Shells, 2nd ed., McGraw-Hill, Kogakusha, 1959, pp. 108-113.

23. Vitásek, E., "Solution of Partial Differential Equations by the Finite-Difference Method," Survay of Applicable Mathematics, Rectorys, K., ed., the M.I.T. Press, 1969, pp. 1109-1124.

24. Wada, A., "Study on Prediction Method of Simulation Analysis for Diffusion of Discharged Warm Water," Proceedings, U.S.-Japan Joint Seminar on Engineering and Environmental Aspects of Heat Disposal, April, 1974.

25. Yatsuzuka, M., Hydraulic Research on Spread of Heated Water from Vertical Multi-Port Diffuser, Thesis Presented to Kyoto University, in Partial Fulfillment of Requirements for the Degree of Doctor of Philosophy, Kyoto University, Kyoto, February, 1975, (in Japanese).

Notations

The following symbols are used in Chapter 2:

$[A] = [a_{np}]$ = state matrix, $(N \times N)$ matrix;

$a_{00} \sim a_{04}$ = coefficients in equilibrium equation;

${}_ja_i$ = area governed by mesh point j;

a_n = area governed by mesh point n:

b = constant in governing equation;

b_l^g = constant in lth constraint;

b_n = constant in nth equilibrium equation;

$C = D_x/\lambda_x^2$, T^{-1};

C_0 = prescribed boundary value;

C_1, C_2 = boundary conditions;

$c_1 \sim c_7$ = coefficients in governing equation;

c_n^ϕ = state-evaluation coefficient = cost coefficient associated with ϕ_n;

c_i^θ = decision-evaluation coefficient = cost coefficient associated with ${}_j\theta_i$;

$[D] = [d_{ni}]$ = decision matrix, $(N \times I)$ matrix;

$D = D_y/\lambda_y^2$, T^{-1};

D_{xk} = diffusion coefficient (D_x, D_y or D_z), L^2T^{-1};

$D.E.$ = differential equation (governing equation);

f = objective function;

$[G_\theta]$ = decision-constraint matrix, $(L_T \times I)$ matrix;

$[G_\phi]$ = state-constraint matrix, $(L_T \times N)$ matrix;

$[g^\theta] = [g_{ii}^\theta]$ = sub-decision-constraint matrix, $(I \times I)$ unit matrix;

$[g^\phi] = [g_{ln}^\phi]$ = sub-state-constraint matrix, $(L \times N)$ matrix;

g = constraint;

h = boundary condition;

I = total number of decision variables (total number of controllable loads);

$i = 1 \sim I$ = decision variable number (controllable load number);

j = mesh point number associated with ith decision variable;

K = heat transfer coefficient at water surface or decay factor of pollutant, T^{-1};

k = index of coordinates;

L = total number of regulated mesh points in water quality requirements;

L_T = total number of constraints in general FDLP Method;

L_x, L_y = lengths in x and y directions of water basin, L;

$l = 1 \sim L$ = water quality requirement number;

$l = 1 \sim L_T$ = constraint number in general FDLP Method;

m = mesh point number associated with lth water quality requirement;

m, n = component numbers in x and y directions in double Fourier series;

N = total number of state variables (total number of mesh points);

$n = 1 \sim N$ = state variable number (mesh point number);

$O(\lambda_{xk}^2)$ = residual term in finite difference approximation ($O(\lambda_x^2)$ or $O(\lambda_y^2)$);

Q^c = decision variable (controllable load);

$\underline{Q}^c$ = lower limit of controllable load;

$\overline{Q}^c$ = upper limit of controllable load;

${}_jQ_i^c$ = ith decision variable (ith controllable load);

${}_j\overline{Q}_i^c$ = upper limit of ith controllable load;

Q^u = uncontrollable load;

Q_n^u = uncontrollable load at mesh point n;

Q_{mn} = known coefficient associated with given load;

$Q(x,y)$ = given load in analytical method;

S = boundary, L or L^2;

U = $v_x/2\lambda_x$, T^{-1};

V = $v_y/2\lambda_y$, T^{-1};

v_k = convective velocity (v_x, v_y or v_z), LT^{-1};

x_k = Cartesian coordinate (x, y or z), L;

Z = objective function;

$\underline{\Theta}$ = lower limit of decision variable;

$\overline{\Theta}$ = upper limit of decision variable;

θ = decision variable;

${}_j\theta_i$ = ith decision variable;

λ_x, λ_y = increments in x and y directions, L;

$\underline{\Phi}$ = lower limit of state variable;

$\overline{\Phi}$ = upper limit of state variable;

${}_m\overline{\Phi}_l$ = lth water quality requirement;

ϕ = state variable (water quality in water basin);

ϕ_n = nth state variable (water quality at mesh point n);

ϕ_{mn} = unknown coefficient associated with state variable in analytical method;

Ω_s = whole domain (whole water basin), L^2 or L^3;

Ω_s^g = subdomain associated with constraints, L^2 or L^3;

Chapter 3

FINITE ELEMENT & LINIEAR PROGRAMMING METHOD IN WATER POLLUTION CONTROL

Summary

Water pollution control is studied by a finite element & linear programming method (FELP Method, or, the F.E. & L.P. Method). FELP Method has been developed by the combined use of finite element method with linear programming in order to solve systems of differential equations with both equality or inequality constraints and an objective function. The applicability of FELP Method is shown through numerical examples of water pollution problems governed by the convective diffusion equation. The method makes it possible to obtain not only the optimal discharges from the various types of outfall to meet water quality requirements, but also the distribution patterns of several water qualities in the water basin simultaneously. A new criterion for selecting the locations of outfalls and the optimal volumes of discharged waste water may be given by the method. The method may become an useful technique for analysis, planning and assessment in environmental and water resources problems. In order to check the results of FELP Method, an analytical method by double Fourier series is developed and described.

3-1. General Concepts

Water pollution control is studied by a finite element & linear programming methd (FELP Method, or, the F.E. & L.P. Method). FELP Method has been developed by the combined use of a finite element method with linear programming in order to solve systems of differential equations with both equality or inequality constraints and an objective function. Such systems are often encountered in various engineering and scientific problems of control and optimal design and, especially, are of interest in environmental and water resources problems. Aguado and Remson (1) have suggested the combined use of finite element method with linear programming in the study of ground water management, in which finite difference method has been used instead of finite element method.

The finite element method, originated in structural mechanics (25), is powerful numerical method for the solution of differential equations because of its generality with respect to geometry and material properties. Moreover, linear programming (4) is one of the most frequently used mathematical methods of oparations research and is widely used in environmental and water resources problems (5, 16, 18, 21). In the development of FELP Method the concepts of the decision variable and state variable are adopted as in Bellman's dynamic programming (2, 8, 15) or Pontryagin's maximum principle (6, 18). FELP Method (9, 10) utilizes the advantages of established numerical techniques of both finite element method and linear programming. The various problems formulated by FELP Method could be solved by the combined use of existing computer codes for finite element method and linear programming.

The applicability of FELP Method is shown through numerical examples of water pollution problems governed by the convective diffusion equation. Many authors have presented the researches on water pollution problems associated with diffusion phenomena (11, 12, 13, 14, 19, 20, 23) and the accomplishments have provided the stimulas and many useful ideas for the application of FELP Method to water pollution control.

The proposed method makes it possible to obtain not only the optimal discharges (the decision variable) from the various types of outfall to meet water quality requirements, but also the distribution patterns of several water quality (the state variable) in the water basin simultaneously. The method become an useful technique for the analysis, planning and assessment in environmental and water resources problems.

In order to check the computations of FELP Method, an analytical method based on double Fourier series is developed and also described.

3-2. Finite Element & Linear Programming Method

3-2-1. Systems of Basic Differential Equations

FELP Method has been developed to solve the following systems of differential equations with both equality or inequality constraints and an objective function. (See Fig. 3-1).

Equilibrium Equations

Governing Equation (in the whole domain Ω_s)

$$D.E.\ \left(x_k,\ \phi,\ \frac{\partial \phi}{\partial x_k},\ .\ .\ .\ ,\ \frac{\partial^n \phi}{\partial x_k^n},\ \theta\right) = 0 \tag{1}$$

Boundary Conditions (on the boundaries S)

$$h\ (X_k,\ \phi) = 0 \tag{2}$$

Constraints (in the sub-domains Ω_s^g)

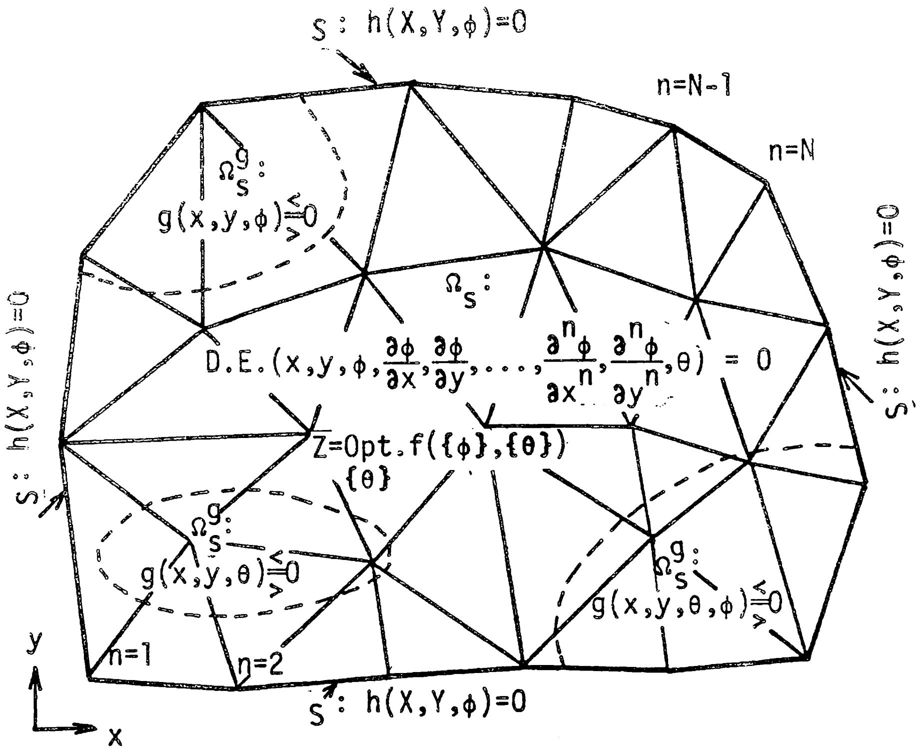
$$g\ (x_k,\ \phi,\ \theta) \lesseqgtr 0 \tag{3}$$

Objective Function (throughout the whole domain Ω_s)

$$Z = \underset{\{\theta\}}{Opt.}\ f\ (\{\phi\},\ \{\theta\}) = \begin{cases} \underset{\{\theta\}}{Max.}\ f\ (\{\phi\},\ \{\theta\}) \\ \underset{\{\theta\}}{Min.}\ f\ (\{\phi\},\ \{\theta\}) \end{cases} \tag{4}$$

in which $\phi = \phi(x_k)$ = the state variable; $\theta = \theta(x_k)$ = the decision variable; x_k = Cartesian coordinate (x, y or z); X_k = Cartesian coordinate of the boundary (X, Y or Z).

In systems governed by two-dimensional second-order differential equations for example, Eq. 1 is expressed as follows:



D.E.: Governing Equation

h : Boundary Condition

g : Constraints

Z : Objective Function

Ω_s: Whole Domain (Whole Water Basin)

S : Boundary

Ω_s^g: Subdomain Associated with Constraints

Fig. 3-1. General Concepts of Finite Element & Linear Programming Method

$$\underbrace{c_1 \frac{\partial^2\phi}{\partial x^2} + c_2 \frac{\partial^2\phi}{\partial x \partial y} + c_3 \frac{\partial^2\phi}{\partial y^2} + c_4 \frac{\partial\phi}{\partial x} + c_5 \frac{\partial\phi}{\partial y} + c_6\phi}_{\phi\text{-terms}} + \underbrace{c_7\theta}_{\theta\text{-term}} + \underbrace{b}_{const} = 0 \tag{1'}$$

The examples of the boundary conditions are as follows:

$$\phi(x=X,\ y=Y) = \Phi_b(X,Y), \quad \frac{\partial\phi}{\partial n}\Big|(x=X,\ y=Y) = 0 \tag{2'}$$

As for the constraints, the following simple inequalities are frequently encountered.

$$\underline{\Phi} \leqq \phi \leqq \overline{\Phi}, \quad or \quad \begin{cases} \phi \geqq \underline{\Phi} \\ \phi \leqq \overline{\Phi} \end{cases}$$

$$\underline{\Theta} \leqq \theta \leqq \overline{\Theta}, \quad or \quad \begin{cases} \theta \geqq \underline{\Theta} \\ \theta \leqq \overline{\Theta} \end{cases} \tag{3'}$$

in which $\underline{\Phi}$ = the lower limit of the state variable; $\overline{\Phi}$ = the upper limit of the state variable; $\underline{\Theta}$ = the lower limit of the decision variable; and $\overline{\Theta}$ = the upper limit of the decision variable.

3-2-2. Formulation of FELP Method

The finite element method is used in order to discretize the above-mentioned systems as systems of linear algebraic equations. (As for the details, see the next section). Then, the following matrix-vector forms of FELP Method are obtained and the application of linear programming is possible.

Equilibrium Equations (N-Eqs.)

$$\underset{(N\times N)}{[\,A\,]}\{\phi_n\} + \underset{(N\times I)}{[\,D\,]}\{{}_j\theta_i\} = \{b_n\} \tag{5}$$

Constraints (L_T-Eqs.)

$$\underset{(L_T\times N)}{[\,G_\phi\,]}\{\phi_n\} + \underset{(L_T\times I)}{[\,G_\theta\,]}\{{}_j\theta_i\} \begin{smallmatrix}\le\\=\\\ge\end{smallmatrix} \{b^g_l\} \tag{6}$$

Nonnegative Conditions

$$\phi_n \geqq 0 \ (n = 1 \sim N), \qquad {}_j\theta_i \geqq 0 \ (i = 1 \sim I) \tag{7}$$

Objective Function

$$Z = \underset{\{{}_j\theta_i\}}{Opt.}\ f\,(\{\phi_n\},\ \{{}_j\theta_i\}) = \underset{\{{}_j\theta_i\}}{Opt.}\ \left(\sum_{n=1}^{N} c^\phi_n\,\phi_n + \sum_{i=1}^{I} c^\theta_i\,{}_j\theta_i\right) \tag{8}$$

in which $[A]$ = the state matrix = the stiffness matrix derived from finite element method and corrected according to the boundary conditions; $[D]$ = the decision matrix, sparse matrix; $[G_\phi]$ = the state-constraint matrix, generally sparse matrix; $[G_\theta]$ = the decision-constraint matrix, generally sparse matrix; ${}_j\theta_i$ = ith decision variable; j = nodal point number associated with ith decision variable; ϕ_n = nth state variable (state variable at the nodal point n); b_n = constant in nth equilibrium equation; b_l^g = constant in lth constarint; c_n^ϕ = state-evaluation coefficient (cost coefficient associated with ϕ_n); c_i^θ = decision-evaluation coefficient (cost coefficient associated with ${}_j\theta_i$); $n = 1 \sim N$ (N: total number of the state variables, i.e., total number of the nodal points in finite elements); $i = 1 \sim I$ (I: total number of the decision variables); and $l = 1 \sim L_T$ (L_T: total number of the constraints).

Therefore, FELP Method is one that optimizes the objective function under the conditions of the equilibrium equations and the constraints. Since all of the variables in linear programming have to be nonnegative because of the limitation in the computational algorithm based on the simplex method (4), the conditions of Eq. 7 are required.

In FELP Method the number of the variables ($N+I$), the number of the equilibrium equations is N, and the number of the constraints is L_T, respectively. In the sense of general linear programming, the equilibrium equations of FELP Method are also the constraints. Thus, FELP Method is a kind of linear programming in which the number of the variables is ($N+I$) and the number of the constraints is ($N+L_T$), respectively.

In FELP Method the solution for the decision variables and the solution for the state variables are obtained simultaneously by the simplex method.

3-3. Water Pollution Control by FELP Method

3-3-1. Systems of Basic Equations in Diffusion-Convection Phenomena

The basic equation systems of two-dimensional diffusion-convection phenomena with constraints and an objective function are as follows:

Equilibrium Equations

Governing Equation (in the whole water basin Ω_s)

$$\underbrace{\sum_{k=1}^{2} \left(\frac{\partial}{\partial x_k} D_{xk} \frac{\partial \phi}{\partial x_k} - v_k \frac{\partial \phi}{\partial x_k}\right) - K\phi}_{\phi\text{-terms}} + \underbrace{Q^c}_{\theta\text{-term}} + \underbrace{Q^u}_{const} = 0 \tag{9}$$

Boundary Conditions (on the boundaries S)

$$\phi(X,Y) = \Phi_b(X,Y) \qquad \text{on } S^1 \tag{10}$$

$$\sum_{k=1}^{2} D_{xk} \frac{\partial \phi}{\partial x_k} l_{xk} = -q - \kappa(\phi - \Phi_a) \qquad \text{on } S^2 \tag{11}$$

Constraints (in the subdomains of the water basin Ω_s^g)

$$\phi \leqq \overline{\Phi} \tag{12}$$

$$Q^c \leqq \overline{Q}^c \tag{13}$$

Nonnegative Conditions

$$\phi \geqq 0, \qquad Q^c \geqq 0 \tag{14}$$

Objective Function (throughout the whole water basin Ω_s)

$$Z = \underset{\{Q^c\}}{Opt.}\ f(\{\phi\}, \{Q^c\}) = \underset{\{Q^c\}}{Max.}\ \Sigma\, Q^c \tag{15}$$

in which ϕ = the state varaible, i.e., water quality (e.g., temperature or concentraition in the water basin); $Q^{c} = \theta$ = the decision variable, i.e., controllable load (heated discharge from multi-port diffusers (24) or pollutant issued from waste outfalls); Q^{u} = uncontrollable load (inevitably or naturally generated source or sink, existing unexcludable discharge); D_{xk} = diffusion coefficient (D_x or D_y); v_k = convective velocity (v_x or v_y); K = heat transfer coefficient at the water surface or decay factor of pollutant; S^1 and S^2 = the parts of the boundary S; Φ_b = prescribed boundary value; l_{xk} = the direction cosine of the outward normal to the boundary (l_x or l_y); q = the intensity of flux per unit length of the boundary; κ = heat transfer coefficient or decay factor of the surrounding boundary; Φ_a = the temperature or concentration of the surrounding boundary; $\overline{\Phi}$ = the upper limit of the state variable (water quality requirement) and $\overline{Q}^{c}$ = the upper limit of the controllable load arising from the conditions in the outfalls.

If D_x and D_y are equal and both q and κ are equal to zero, a well-known condition applicable to non-conductive boundaries is obtained, that is $\partial\phi/\partial n = 0$.

Although, the objective function may be composed of the controllable loads and water quality distribution in general, the maximization of the total of the controllable loads is sought in the numerical examples for simplicity. From the view point of the assimilation capacity of the environment, such an objective function gives us the upper limit of the total acceptable load in the water basin.

As for the constraints, although only the upper limits of the state variable and the decision variable (controllable load) are imposed in the above systems, the lower limit of the decision variable may occasionally become necessary with respect to the conditions of the problems. Such a condition may occasionally arise from the hydraulic conditions in the outfalls or from the treatment efficiencies in plants.

$$Q^c \geq \underline{Q}^c \tag{16}$$

The formulation presented above is concerned with a two-dimensional scheme which is suitable for large bodies of water such as lakes and ocean. The scheme may be easily extended to three-dimensional problems in a similar manner.

3-3-2. Formulation of FELP Method in Water Pollution Control

The application of FELP Method to the systems of the diffusion-convection phenomena mentioned above yields the following matrix-vector forms.

Equilibrium Equations (N-Eqs.)

$$\underset{(N\times N)}{[\ A\ ]}\{\phi_n\} + \underset{(N\times I)}{[\ D\ ]}\{{}_jQ_i^c\} = \{Q_n^u\} \tag{17}$$

Constraints ($(L_T = L+I)$-Eqs.)

$$\underset{(L\times N)}{[\ g^\phi\ ]}\{\phi_n\} \leqq \{{}_m\bar{\Phi}_l\} \tag{18}$$

$$\underset{(I\times I)}{[\ g^\theta\ ]}\{{}_jQ_i^c\} \leqq \{{}_j\bar{Q}_i^c\} \tag{19}$$

Nonnegative Conditions

$$\phi_n \geqq 0\ (n = 1 \sim N), \qquad {}_jQ_i^c \geqq 0\ (i = 1 \sim I) \tag{20}$$

Objective Function

$$Z = \underset{\{{}_jQ_i^c\}}{Opt.}\ f\ (\{\phi_n\},\ \{{}_jQ_i^c\}) = \underset{\{{}_jQ_i^c\}}{Opt.}\ (\sum_{n=1}^{N} c_n^\phi\ \phi_n + \sum_{i=1}^{I} c_i^\theta\ {}_jQ_i^c)$$

$$\approx \underset{\{{}_jQ_i^c\}}{Max.}\ \sum_{i=1}^{I} {}_jQ_i^c \tag{21}$$

in which [A] = the state matrix = the stiffness matrix; [D] = the decision matrix; $[g^\phi]$ = the sub-state-constraint matrix; $[g^\theta]$ = the sub-decision-constraint matrix, unit matrix; ${}_jQ_i^c = {}_j\theta_i$ = ith controllable load (ith decision variable); j =nodal point number fitted for the location of ith controllable load; ϕ_n = water quality at the nodal point n (nth state variable); Q_n^u = uncontrollable load at the nodal point n; ${}_j\overline{Q}_i^c$ = upper limit of ith controllable load; ${}_m\overline{\Phi}_l$ = lth water quality requirement; m = regulated nodal point number in lth water quality requirement; $i = 1 \sim I$ (I: total number of the controllable loads, i.e., total number of nodal points fitted for the locations of the outfalls of the controllable loads); $n = 1 \sim N$ (N: total number of the state variables, total number of the nodal points); $l = 1 \sim L$ (L: total number of the regulated nodal points in water quality requirements); and $L_T = L + I$ = total number of the constraints.

In the consideration of the loads, it should be noted that in FELP Method, as in finite element method, concentrated loads are fundamental and distributed loads are displaced by the equivalent concentrated loads. Therefore, in FELP Method all of the loads expressed in ${}_jQ_i^c$ and Q_n^u are concentrated loads at the nodal points.

In the maximization problem of total of the controllable loads, all of the state-evaluation coefficients (cost coefficients associated with $\{c_n^\phi\}$) are equal to zero and all of the decision-evaluation coefficients (cost coefficients associated with $\{c_i^\theta\}$) are equal to '1' as shown in Eq. 21.

Eqs. 18 and 19 are equal to the following equations.

$$\phi_m \leqq {}_m\overline{\Phi}_l \quad (l = 1 \sim L) \tag{18'}$$

$$ {}_jQ_i^c \leqq {}_j\overline{Q}_i^c \quad (i = 1 \sim I) \tag{19'}$$

As for the details of the matrices [A], [D], $[g^\phi]$ and $[g^\theta]$, see the next section and Eqs. 34-36.

3-3-3. Discretization by Finite Element Method

Finite element method, originated in structural mechanics, has been extended to many other physical phenomena because of its generality with respect to geometry and material properties. Finite element analysis has been extended to diffusion phenomena recently (3, 17). Various formulation techniques of the finite element method have been developed. In this research Galerkin finite element method is used because of its independency of variational principle. As for the details of Galerkin finite element method, one may follow Zienkievicz (25).

The water basin to be analyzed is divided into small regions called finite elements. (See Fig. 3-1). Two kinds of triangular elements shown in Figs. 3-2(a) and 3-2(b) are adopted in this research. The central idea common to all varieties of the finite element method is the description of the variation of unknown (the state variable ϕ for the present instance) by shape functions in each element. The description is given by the following approximation.

$$\phi = [N]\{\phi\}^e = \sum_{r=1}^{R} N_r \phi_r \tag{22}$$

in which $[N] = [N_1 \,..\, N_r \,..\, N_R]$ = the usual shape functions expressed by area coordinates (L_1, L_2, L_3), (see Notes); $\{\phi\}^e$ = listing of nodal values for paticular element; R =number of the nodal points in an element = 3 (in Fig. 3-2(a)) or 6 (in Fig. 3-2(b)); and ϕ_r discrete nodal representation.

Using the weighted residual process (25) for the diffusion convection equation (Eq. 9), we obtain the following equation.

$$\iint_{\partial_s} W_n \left[\sum_{k=1}^{2} \frac{\partial}{\partial x_k} \left(D_{xk} \frac{\partial \phi}{\partial x_k}\right) - \sum_{k=1}^{2} v_k \frac{\partial \phi}{\partial x_k} - K\,\phi + Q^c + Q^u \right] dx\, dy = 0 \tag{23}$$

in which W_n = weighting function.

Integrating the first term in Eq. 23 by parts, we obtain the following nth equation in the N-equilibrium equations.

$$\int_{S^2} W_n \sum_{k=1}^{2} \frac{\partial \phi}{\partial x_k} l_{xk} \, dS^2$$

$$- \iint_{\Omega_s} \left[\sum_{k=1}^{2} D_{xk} \frac{\partial W_n}{\partial x_k} \frac{\partial \phi}{\partial x_k} + \sum_{k=1}^{2} v_k W_n \frac{\partial \phi}{\partial x_k} + K W_n \phi - Q^c W_n - Q^u W_n \right] dx \, dy = 0 \quad (24)$$

Using the expression of the boundary conditions (Eq. 11), the first term in Eq. 24 is rewritten as follows:

$$\int_{S^2} W_n \sum_{k=1}^{2} \frac{\partial \phi}{\partial x_k} l_{xk} \, dS^2 = - \int_{S^2} W_n \, q \, dS^2 - \int_{S^2} W_n \, \kappa \, (\phi - \phi_a) \, dS^2 \quad (25)$$

In Galerkin finite element method the shape functions are used as the weighting functions.

$$W_n = N_n \quad (26)$$

Inserting Eqs. 22, 25 and 26 into Eq. 24, the element contribution to the integrals in Eq. 24 can be written as follows:

$$- \sum_{s=1}^{R} a_{rs}^e \phi_s + q_r^c + q_r^u = 0 \quad (r = 1 \sim R) \quad (27)$$

with

$$a_{rs}^e = \sum_{k=1}^{2} D_{xk} \int_\Delta \frac{\partial N_r}{\partial x_k} \frac{\partial N_s}{\partial x_k} d\Delta + \sum_{k=1}^{2} v_k \int_\Delta N_r \frac{\partial N_s}{\partial x_k} d\Delta$$

$$+ K \int_\Delta N_r N_s \, d\Delta + \kappa \int_{s^2} N_r N_s \, ds^2 \quad (r, s = 1 \sim R) \quad (28)$$

$$q_r^c = Q^c \int_\Delta N_r \, d\Delta \quad (r = 1 \sim R) \quad (29)$$

$$q_r^u = Q^u \int_\Delta N_r \, d\Delta - q \int_{s^2} N_r \, ds^2 + \kappa \, \Phi_q \int_{s^2} N_r \, ds^2 \quad (r = 1 \sim R) \tag{30}$$

in which s = summation subscript; a_{rs}^e = element (factor) of element stiffness matrix; q_r^c = equivalent nodal controllable load at the nodal point r; q_r^u = equivalent nodal uncontrollable load at the nodal point r; Δ = area of each triangular element; and s^2 = segment of the boundary S^2.

Summation of the contribution from the all elements yields the following nth equilibrium equation.

$$\sum_{p=1}^{N} a_{np} \, \phi_p - Q_n^c = Q_n^u \quad (n = 1 \sim N) \tag{31}$$

with

$$a_{np} = \Sigma \, a_{np}^e, \quad Q_n^c = \Sigma \, q_n^c, \quad Q_n^u = \Sigma \, q_n^u \tag{32}$$

in which p = summation of the global stiffness matrix; and Q_n^c = controllable load at the nodal point n.

The notation Σ in Eq. 32 means the summation over all the elements. In order to reduce the number of the decision variables from N to I, Q_n^c in Eq. 31 should be dropped at the nodal point where the outfall for the controllable load does not exist. Then, in Eq. 17 $_jQ_i^c$ is used instead of Q_n^c. The decision matrix in Eq. 17 is composed of zero elements with the exceptions of '-1'in I elements whose row number is j and whose column number is i (see. Eq. 34).

On the boundary S^1 with the prescribed boundary value of $\phi = \Phi_b$ (Eq. 10), the equilibrium equation is corrected as follows:

$$a_{nn} \, \phi_n = \Phi_b, \; or \quad 1 \times \phi_n = \Phi_b \tag{33}$$

Thus the descretized equilibrium equations (Eq. 17) are obtained.

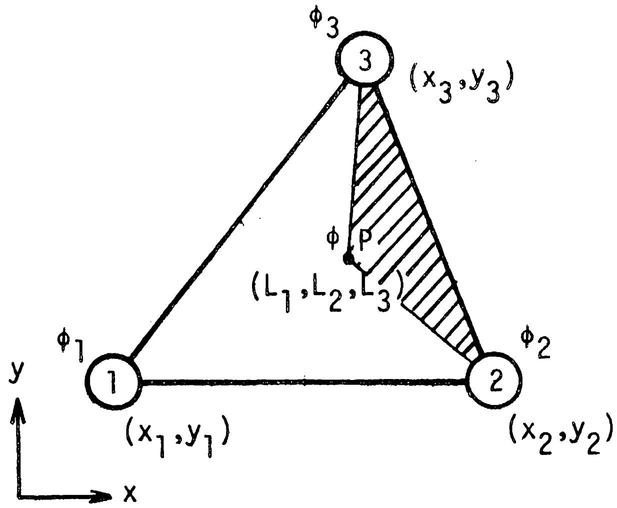


Fig. 3-2(a). Triangular Element with Three Nodal Points

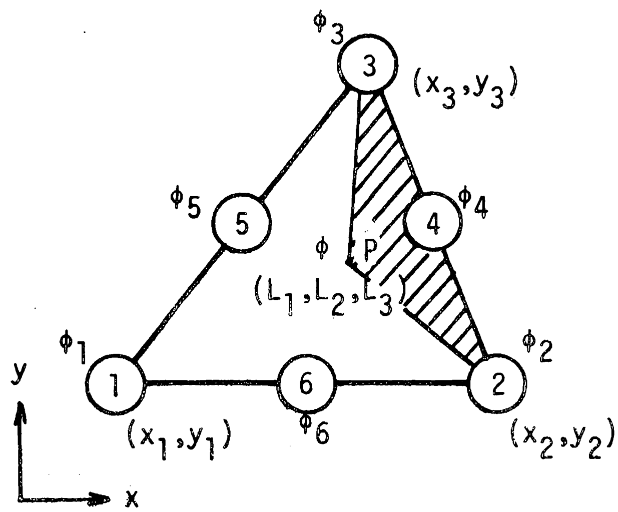


Fig. 3-2(b). Triangular Element with Six Nodal Points

3-4. Explanation by a Simple Model

In order to clarify the features of FELP Method, the systems of the governing equations are written down for a simple square model basin in a specific form with non-conductive boundaries, or $\partial\phi/\partial n = 0$ on the four surrounding boundaries as shown in Fig. 3-3. The obtained matrix-vector forms of FELP Method are as follows:

$$
\begin{array}{c}
\\ 1\\2\\3\\4\\5\\6\\7\\8\\9\\1\\2\\3\\4\\5\\6\\7\\8
\end{array}
\begin{array}{ccccccccc|ccc}
1&2&3&4&5&6&7&8&9&1&2&3\\
\hline
&&&&&&&&&&&\\
&&&&&&&&&-1&&\\
&&&&&&&&&&&\\
&&&&A&&&&&&&\\
&&&&(9\times9)&&&&&&-1&\\
&&&&&&&&&&D&\\
&&&&&&&&&&(9\times3)&\\
&&&&&&&&&&&-1\\
&&&&&&&&&&&\\
\hline
&&&1&&&&&&&&\\
&&&&&1&&&&&\text{Zero}&\\
&&&&g^{\phi}&&1&&&&\text{Matrix}&\\
&&&&(5\times9)&&&1&&&(5\times3)&\\
&&&&G_{\phi}&&&&1&&G_{\theta}&\\
\hdashline
&&&&(8\times9)&&&&&1&(8\times3)&\\
&&&&\text{Zero Matrix}&&&&&&1\ g^{\theta}&\\
&&&&(3\times9)&&&&&&(3\times3)&1
\end{array}
\begin{Bmatrix}
\phi_1\\ \phi_2\\ \phi_3\\ \phi_4\\ \phi_5\\ \phi_6\\ \phi_7\\ \phi_8\\ \phi_9\\ {}_2Q^c_1\\ {}_5Q^c_2\\ {}_8Q^c_3
\end{Bmatrix}
\begin{array}{ll}
= Q^u_1 & (= 3.0)\\
= Q^u_2 & (= 0)\\
= Q^u_3 & (= 3.0)\\
= Q^u_4 & (= 0)\\
= Q^u_5 & (= 0)\\
= Q^u_6 & (= 0)\\
= Q^u_7 & (= 0)\\
= Q^u_8 & (= 0)\\
= Q^u_9 & (= 0)\\
\leqq {}_4\overline{\Phi}_1 & (= 3.0)\\
\leqq {}_6\overline{\Phi}_2 & (= 3.0)\\
\leqq {}_7\overline{\Phi}_3 & (= 3.0)\\
\leqq {}_8\overline{\Phi}_4 & (= 3.0)\\
\leqq {}_9\overline{\Phi}_5 & (= 3.0)\\
\leqq {}_2\overline{Q}^c_1 & (= 30.0)\\
\leqq {}_5\overline{Q}^c_2 & (= 30.0)\\
\leqq {}_8\overline{Q}^c_3 & (= 30.0)
\end{array}
$$

(34) [rows 1–9]

(35) [rows 1–5 of the lower block]

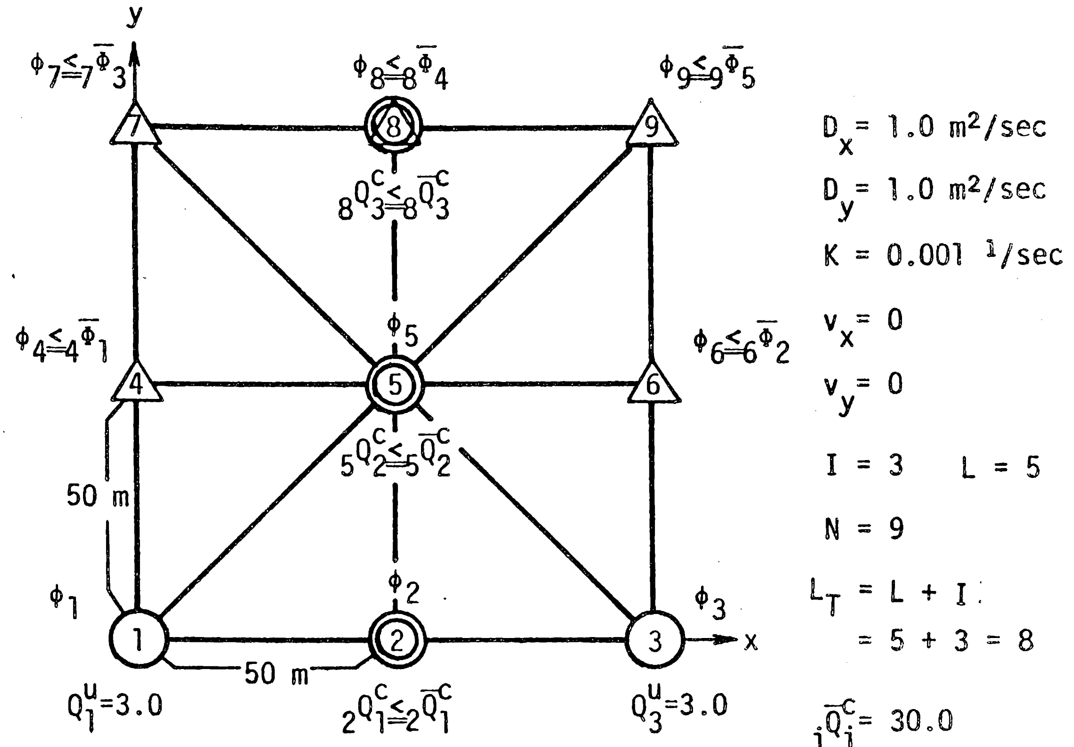
(36) [rows 6–8 of the lower block]

$$\phi_n \geqq 0 \ (n = 1 \sim 9), \qquad {}_jQ^c_i \geqq 0 \ (i = 1 \sim 3) \tag{37}$$

$$Z = Max. \sum_{i=1}^{3} {}_{j}Q_{i}^{c} = Max. \; ({}_{2}Q_{1}^{c} + {}_{5}Q_{2}^{c} + {}_{8}Q_{3}^{c}) \tag{38}$$

The units of the water quality and the loads are not described, because the method is applicable to general physical problems irrespective of the variables considered.

Computed results for the controllable loads $\{{}_{j}Q_{i}^{c}\}$, water qualities $\{\phi_n\}$ and objective function Z are shown in Fig. 3-4. It should be noted that the distribution pattern of the water quality in Fig. 3-4 arises from the resultant loads composed of the obtained controllable loads $\{{}_{j}Q_{i}^{c}\}$ and the given uncontrollable loads $\{Q_{n}^{u}\}$.



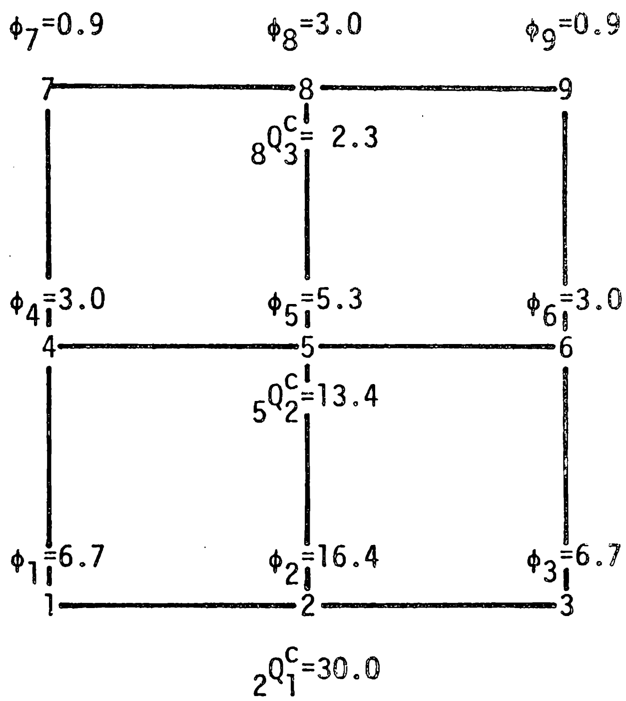
$\partial\phi/\partial n = 0$ (On the Four Boundaries)

◎: Nodal Point for Controllable Load

○: Nodal Point for Uncontrollable Load

△: Regulated Nodal Point in Water Quality Requirement ($_m\bar{\Phi}_l = 3.0,\ l = 1 \sim 5$)

Fig. 3-3. Input Data on FELP Method in a Simple Model Basin



$$Z = {}_2Q_1^c + {}_5Q_2^c + {}_8Q_3^c$$
$$= 30.0 + 13.4 + 2.3 = 45.7$$

Fig. 3-4. Results of FELP Method in a Simple Model Basin

3-5. Numerical Examples in a Model Basin

Numerical examples of FELP Method are conducted in a rectangular model basin as shown in Fig. 3-5. Two runs are done under the different conditions as shown in Table 3-1. However, the input data shown in Fig. 3-5 and those described below are the same in both runs.

Consider the case in which there will be further increases of steady waste discharges over the exisiting steady waste discharges. The existing steady waste discharges, or the uncontrollable loads, are, Q^u_{43}, Q^u_{60}, Q^u_{77} and Q^u_{94}, respectively. They are known loads and their magnitude is equally 30.0. Further increase of the outfalls for the waste discharges due to the increase of power of the plants is planned. The locations of the outfalls are planned at the six nodal points, or, 42, 59, 76, 93, 110 and 127. The waste discharges issued from these six outfalls are the controllable loads to be solved, or, ${}_{42}Q^c_1$, ${}_{59}Q^c_2$, ${}_{76}Q^c_3$, ${}_{93}Q^c_4$, ${}_{110}Q^c_5$ and ${}_{127}Q^c_6$. The imposed upper limit of the controllable loads ${}_j\overline{Q}^c_i$ is equally 30.0 $(i = 1 \sim 6)$. Twenty eight nodal points are regulated in the same water quality requirement, or, ${}_m\overline{\Phi}_l = 3.0$ $(l = 1 \sim 28)$.

we would like to know which part of the increase of waste discharges should be allocated to these six outfalls to meet the water quality requirement and also to know the maximum of the acceptable volume of the increase of waste discharges. The maximum should be known from the standpoint of environmental assessment and is given by the total of the controllable loads, or the objective function.

The solutions for the controllable loads and the objective function in both runs are shown in Table 3-2. The sets of the solution $\{{}_jQ^c_i\}$ give us the optimal locations for outfalls shown by the nodal point number j and the optimal volumes shown by the $\{{}_jQ_i^c\}$ themselves.

The distribution patterns of the water quality obtained by the resultant loads composed of the controllable loads $\{{}_jQ_i^c\}$ and the uncontrollable loads $\{Q_n^u\}$ are also shown in Figs. 3-6(a) and 3-6(b).

The units of the water quality and the loads are not attached in order to show that the method is applicable to various water qualities.

Table 3-1. Input Data on Current Speed and Boundary Conditions

Conditions (1)		Run 1 (2)	Run 2 (3)
Current Speed		$v_x = 0$, $v_y = 0$	$v_x = 0.03$ m/sec, $v_y = 0$
Boundary Conditions	Boundary 1	$\partial\phi/\partial y = 0$	$\partial\phi/\partial y = 0$
	Boundary 2	$\partial\phi/\partial x = 0$	$\phi = 0.1$
	Boundary 3	$\partial\phi/\partial x = 0$	$\phi = 1.5$
	Boundary 4	$\partial\phi/\partial y = 0$	$\partial\phi/\partial y = 0$

Table 3-2. Controllable Loads and Objective Function

Controllable Load Number i (1)	Associated Nodal Point j (2)	Run 1 Controllable Load (Concentrated Load) ${}_jQ_i^c$ (3)	Run 2 Controllable Load (Concentrated Load) ${}_jQ_i^c$ (4)
1	42	0.0	30.0
2	59	0.0	7.5
3	76	8.1	0.0
4	93	30.0	30.0
5	110	30.0	30.0
6	127	30.0	30.0
Total (Objective Function)		Z = 98.1	Z = 127.5

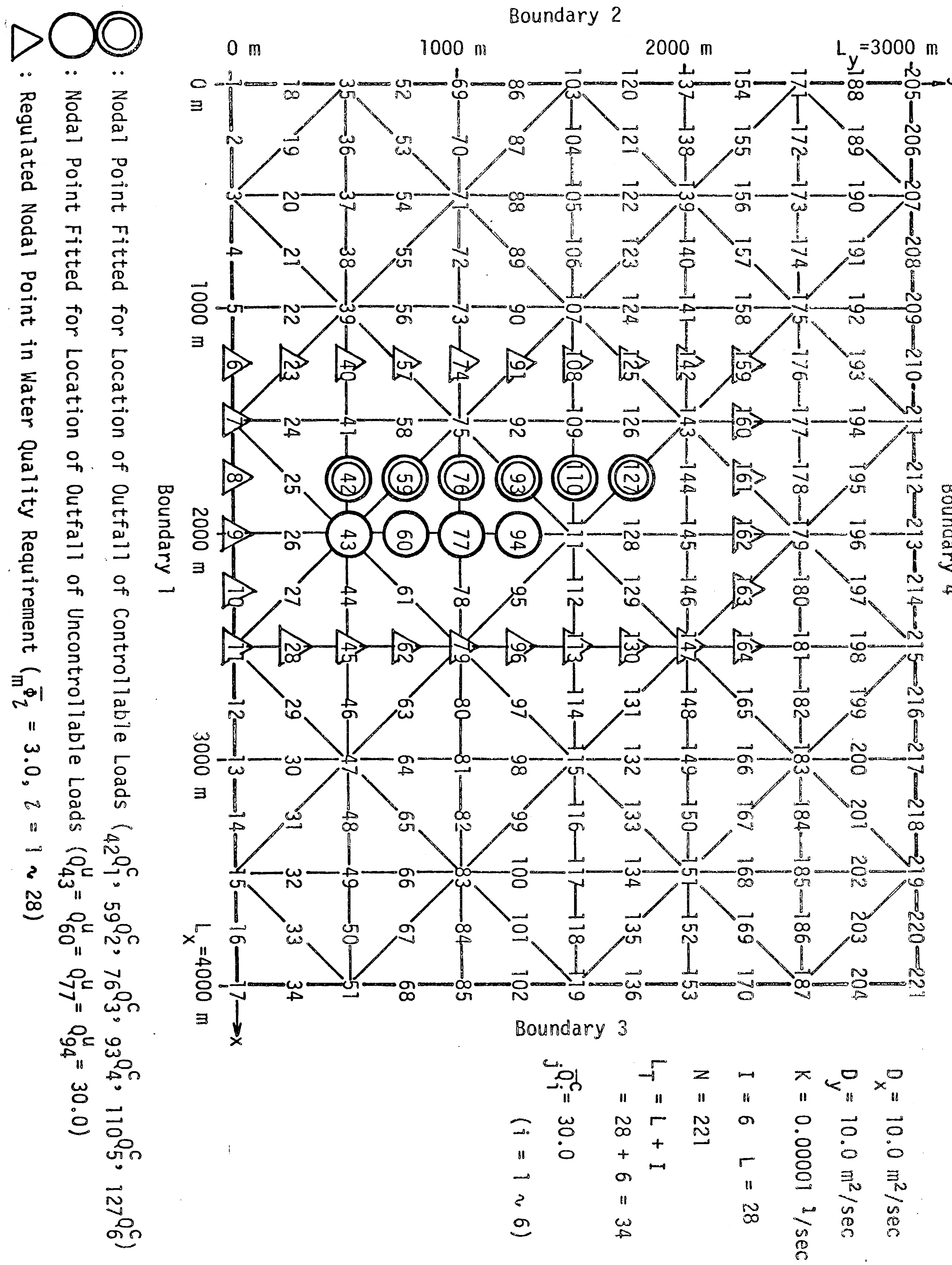


Fig. 3-5. Input Data on FELP Method in a Rectangular Model Basin

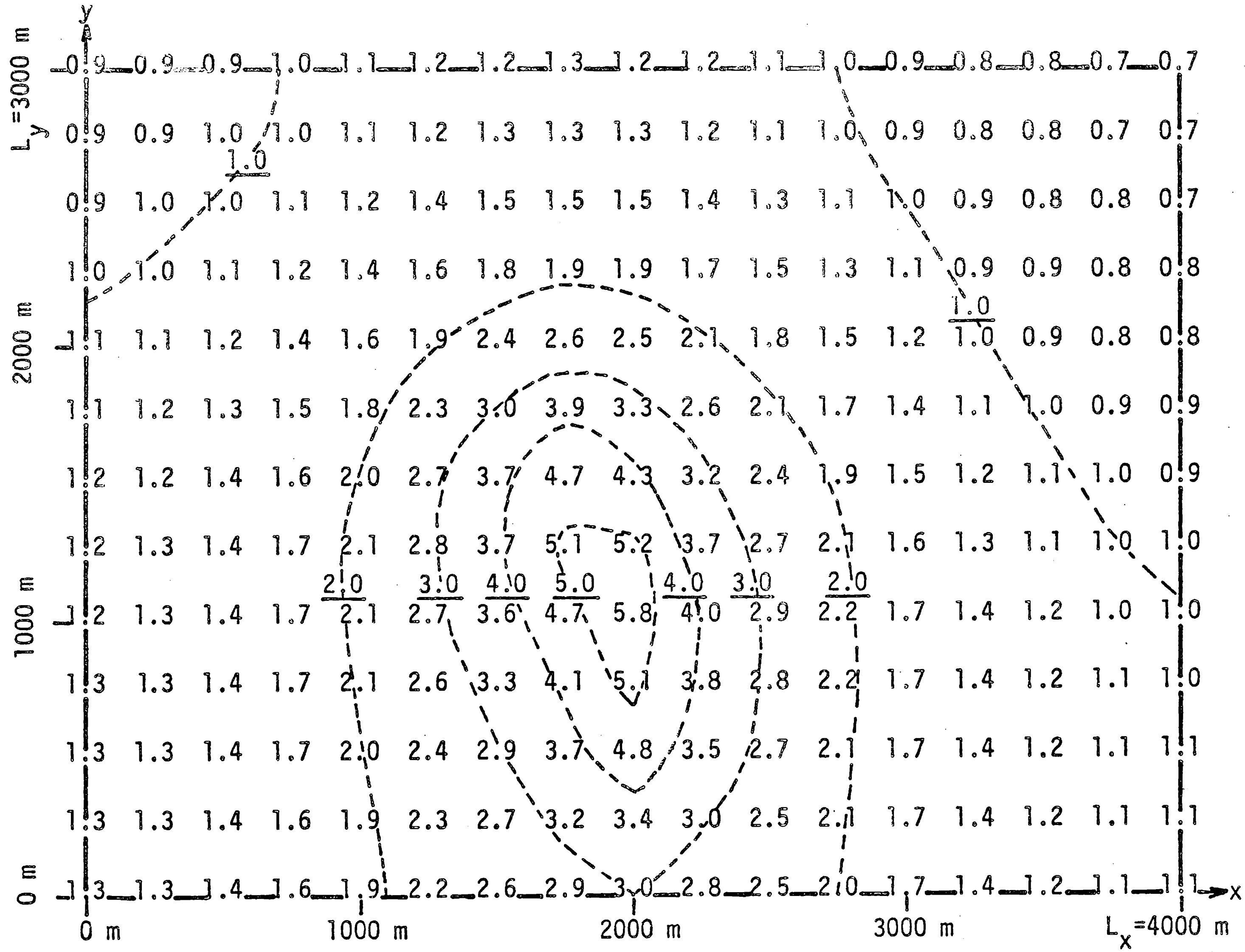


Fig. 3-6(a). Results of FELP Method in Run 1 (Distribution Pattern of Water Quality)

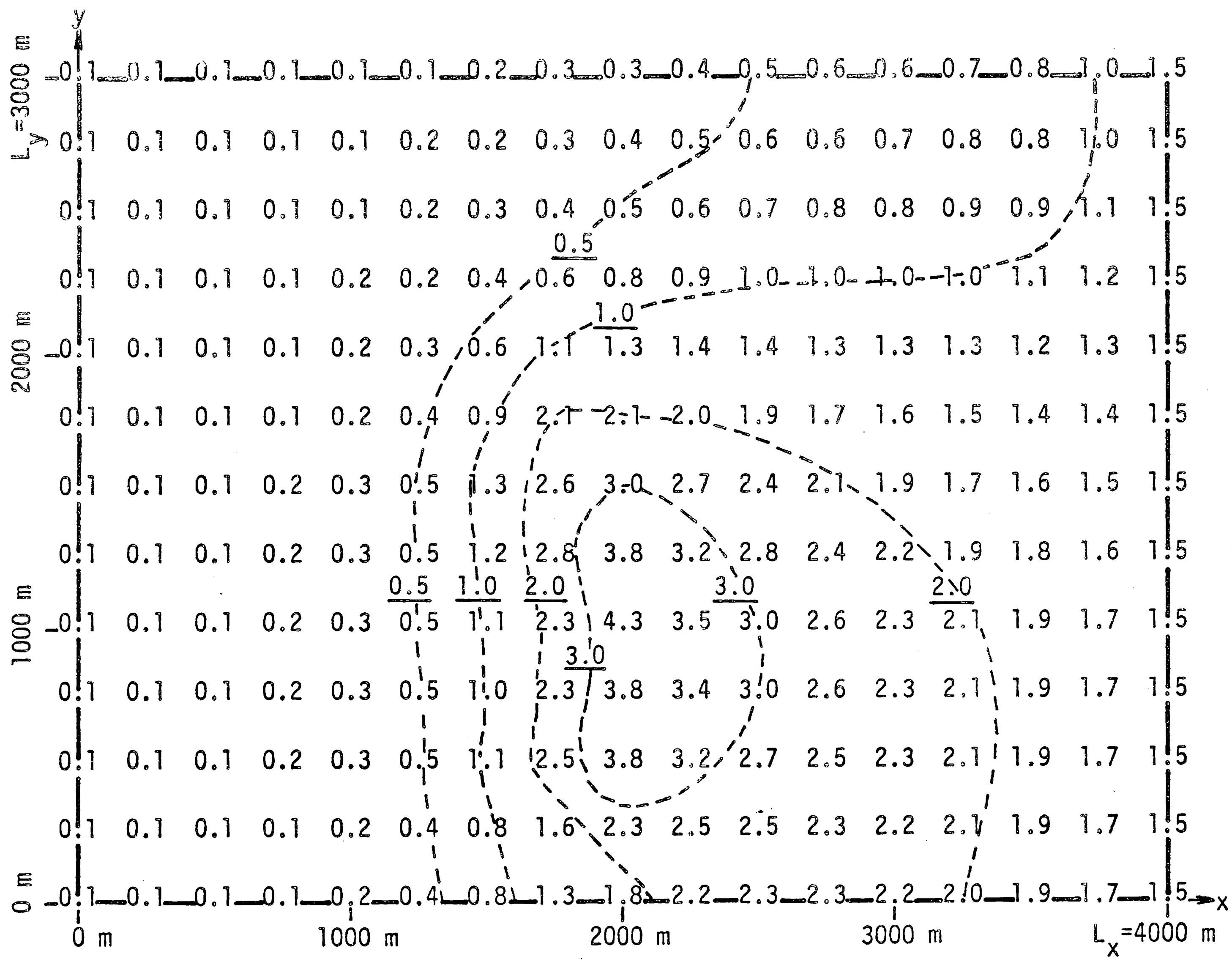


Fig. 3-6(b). Results of FELP Method in Run 2 (Distribution Pattern of Water Quality)

3-6. Comparison with Analytical Method

3-6-1. Analytical Method by Double Fourier Series

An analytical method by Fourier series (7) is considered in order to check the computations of FELP Method.

Consider the following diffusion equation with non-conductive boundaries.

$$D_x \frac{\partial^2 \phi}{\partial x^2} + D_y \frac{\partial^2 \phi}{\partial y^2} - K \phi + Q(x,y) = 0 \tag{39}$$

It is desirable to have a solution of Eq. 39 for a given load $Q(x,y)$. Such a solution can be found if the load is distributed according to the following formula:

$$Q(x,y) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} Q_{mn} \cos \frac{m\pi}{L_x}x \cos \frac{n\pi}{L_y}y \tag{40}$$

where Q_{mn} is a constant and expressed in the following equation.

$$Q_{mn} = \frac{2}{L_x} \int_0^{L_x} \frac{2}{L_y} \int_0^{L_y} Q(x,y) \cos \frac{m\pi}{L_x}x \cos \frac{n\pi}{L_y}y \, dy \, dx \tag{41}$$

$$(m = 0, 1, 2, \ldots, \infty); \quad (n = 0, 1, 2, \ldots, \infty)$$

Such a technique in which the load is displaced by double Fourier series so as to satisfy the given boundary conditions is used in structural analyses. The first solution of the problem of bending of simply supported rectangular plates and the use for this purpose of double Fourier series are due to Navier, who presented a paper on this subject to French Academy in 1820 (22).

Introducing Eq. 40 into Eq. 39, it may be seen that there exists a solution in the following form

$$\phi = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \phi_{mn} \cos \frac{m\pi}{L_x}x \cos \frac{n\pi}{L_y}y \tag{42}$$

with unknown constant ϕ_{mn}.

Substituting Eqs. 40 and 42 into Eq. 39 and dropping the trigonometric factors, the following expression for ϕ_{mn} is obtained.

$$\phi_{mn} = \frac{Q_{mn}}{D_x \left(\frac{m\pi}{L_x}\right)^2 + D_y \left(\frac{n\pi}{L_y}\right)^2 + K} \tag{43}$$

Substitution of the above expression into Eq. 42 yields the following solution.

$$\phi = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \frac{Q_{mn}}{D_x \left(\frac{m\pi}{L_x}\right)^2 + D_y \left(\frac{n\pi}{L_y}\right)^2 + K} \cos \frac{m\pi}{L_x}x \cos \frac{n\pi}{L_y}y \tag{44}$$

3-6-2. Check Analysis of Run 1 by Analytical Method

The distribution pattern of the water quality in Run 1 is obtained by FELP Method in the previous section as shown in Fig. 3-6(a) is compared with that obtained by the analytical method mentioned above.

The input data used with the analytical method are shown in Fig. 3-7. The input loads $Q(x,y)$ in the analytical method are the resultant loads composed of the given uncontrollable loads $\{Q_n^u\}$ and the obtained controllable loads $\{{}_jQ_i^c\}$ in Run 1. The maximum number for both m and n adopted in the check analysis is *100* and double precision is used.

The results obtained by the analytical method are shown in Fig. 3-8 and show good agreement with those obtained by FELP Method.

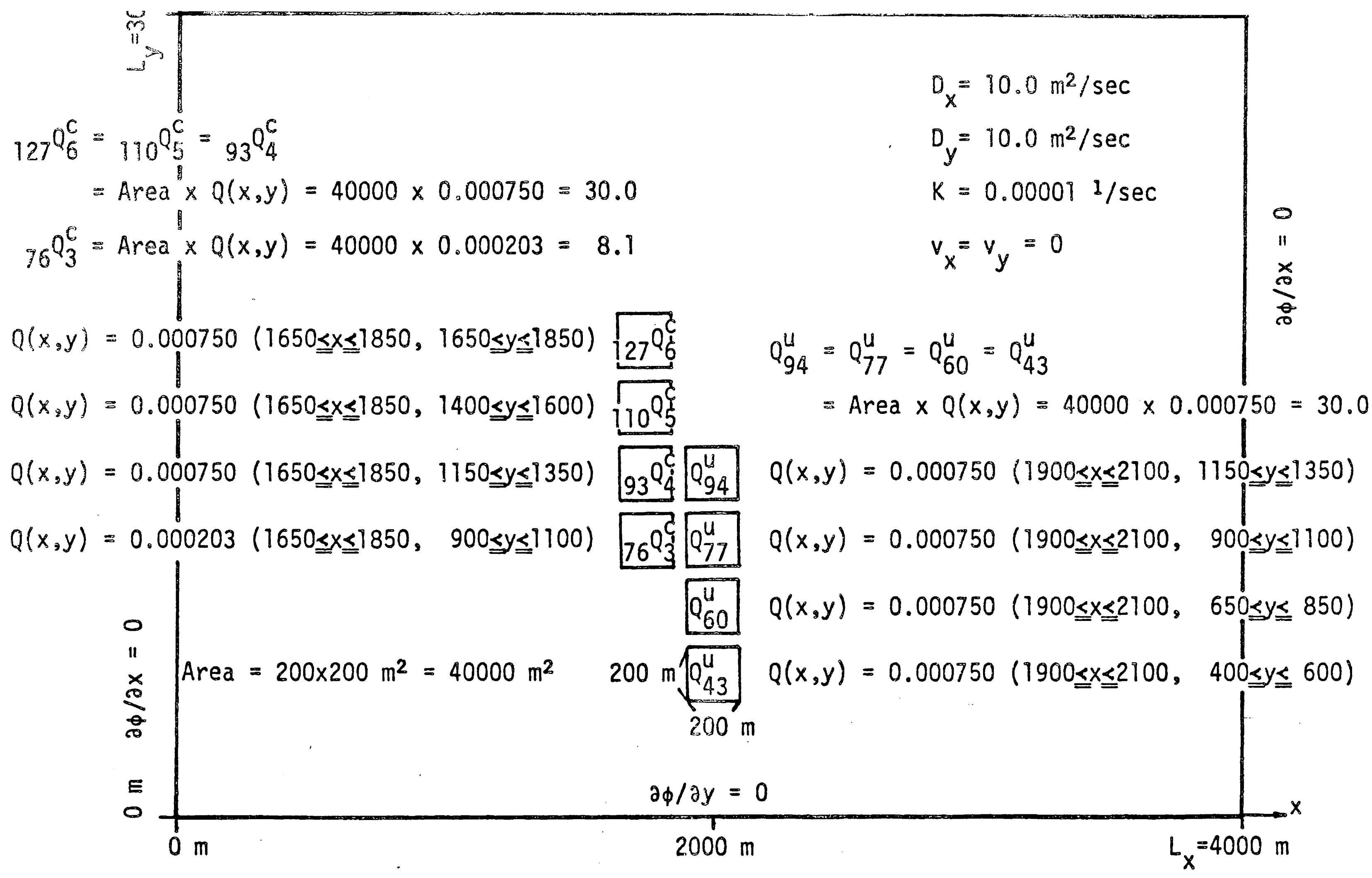


Fig. 3-7. Input Data for Check of Run 1 by Analytical Method

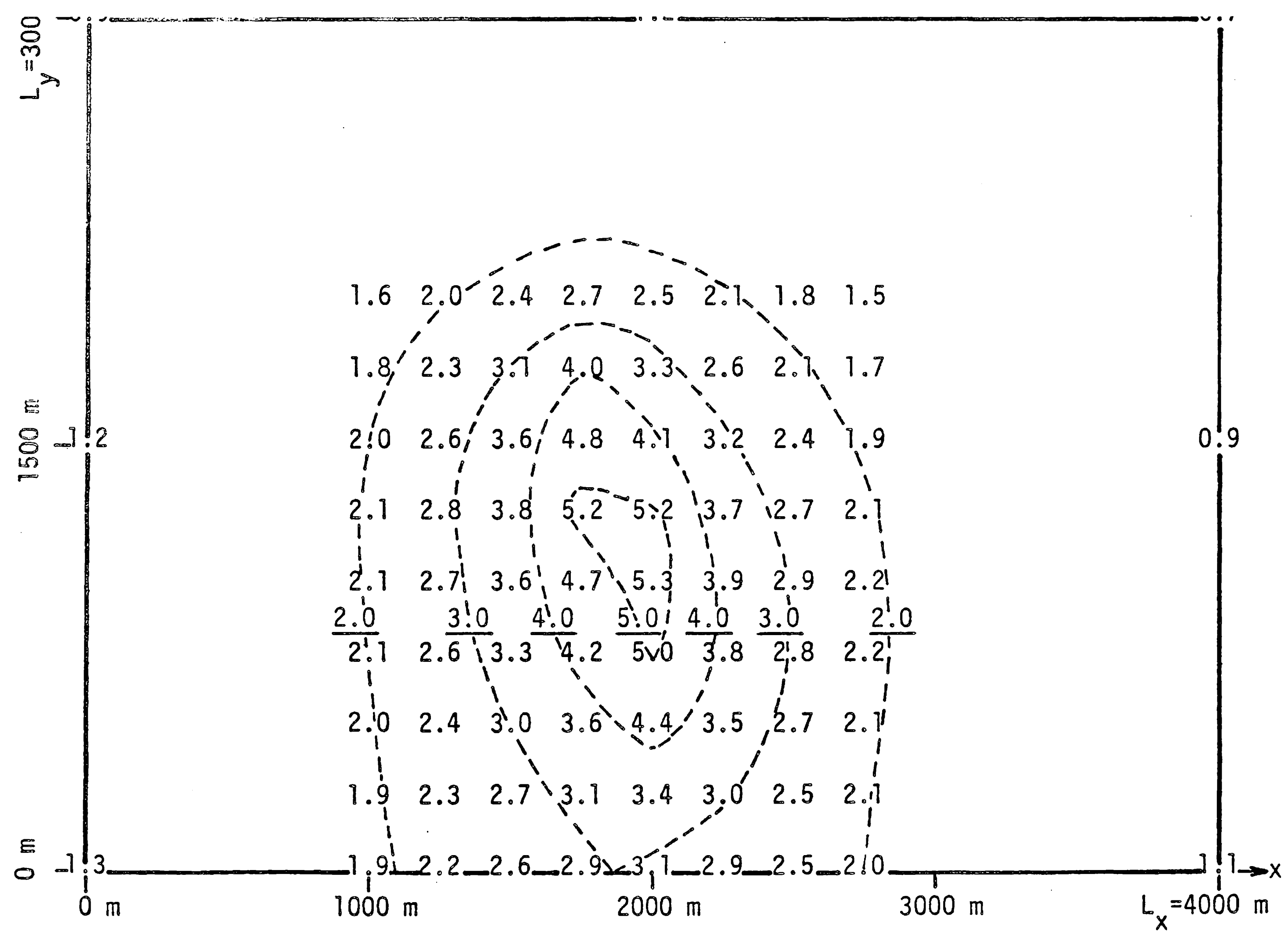


Fig. 3-8. Results of Check of Run 1 by Analytical Method (Distribution Pattern of Water Quality)

3-7. Concluding Remarks

Finite element & linear programming method (FELP Method, or, the F.E. & L.P. Metod) for water pollution control was described. Some numerical examples in model basins were also presented. The computations of FELP Method were checked by an analytical method based on double Fourier series.

A new criterion for selecting the locations of outfalls and the optimal volumes of discharged waste water may be given by FELP Method. The tractability in both the boundary conditions and the equality or inequality constraints makes sure that the method becomes powerful technique for several new types of boundary value problems.

Most practical applications of linear programming make use of the degital computer and existing computer codes. However, in order to save computer time and memory, an efficient computational algorithm of FELP Method has been developed by taking note of the fact that the method has special structures. The details are presented in Chapter 4.

In a manner similar to FELP Method, finite difference & linear programming method (FDLP Method, or, the F.D. & L.P. Method) has been developed by the combined use of the finite difference method with linear programming. (See. Chapter 2). Aguado and Remson made a pioneering research associated with FDLP Method in the field of ground water management (1).

Finally, the problems to be attacked from now on in the applications of FELP Method should be mentioned. Extension of the method to the time domain is necessary to solve transient problems. As for the extension of FELP Method to the time domain (transient finite element & linear programming method, or, T. FELP Method), see Chapter 5. The related methods such as finite element & non-linear programming method and finite element & integer programming method could be developed. The developments certainly make it possible to solve more complicated problems. Review and further progress of finite element method (the basis of FELP Method) in fluid mechanics will widen the applicability of the proposed method. Comparison with other analytical methods, and experiments and field data should be extended to make the applicability of the method wider.

The computational work in this chapter was performed by using the computer center of the University of Tokyo and its program library for the simplex method "HI/TC/LP02 (made by K. Ikura and revised by Hitachi Co. Ltd.)".

References

1. Aguado, E. and Remson, I., "Ground-Water Hydraulics in Aquifer Management," Journal of the Hydraulics Division, ASCE, Vol. 100, No. HY1, January, 1974, pp. 103-118.
2. Bellman, R., "Dynamic Programming," Princeton University Press, 1975, p. 81.
3. Bruch, JR.C.J. and Zyvoloski, G., "Transient Two-Dimensional Heat Conduction Problems Solved by the Finite Element Method," International Journal for Numerical Methods in Engineering, Vol. 8, 1974, pp. 481-494.
4. Dantzig, B.G., "Linear Programming and Extensions," Princeton University Press, 1963, pp. 94-119.
5. Deininger, R.A., "Water Quality Management: The Planning of Economically Optimal Pollution Control Systems," Thesis Presented to Northwestern University, in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy, June, 1965.
6. Fan, L.T., "The Continuous Maximum Principle," John Wiley & Sons, 1966, p. 2, p 12.
7. Fourier, "Propagation de la Chaleur dans un Solide Rectangulaire Infini," Théorie Analytique de la Chaleur, Œbres de Fourier, Darboux, G., ed., Tome Premier, Gauthier-Villars et Fils, Imprimeurs-Libraires, Paris, 1887, pp. 141-238.
8. Futagami, T., "Dynamic Programming for a Sewage Treatment System," Proceedings, 5th International Water Pollution Research Conference, Jenkins, H.S., ed., Pergamon Press Ltd., 1971, pp. II-21/1-II-21/12.
9. Futagami, T., "Development of Finite Element & Linear Programming

Method and Its Application to Water Pollution Control Problems," Proceedings, 19th Conference on Hydraulic Research, JSCE, February, 1975, pp. 133-138, (in Japanese).

10. Futagami, T., "Finite Element & Linear Programming Method and Water Pollution Control," Proceedings, 16th Congress of the International Association for Hydraulic Research, July-August, 1975, C7, pp. 54-61.

11. Hayashi, T. and Shuto, N., "Diffusion of Warm Water Jets Discharged Horizontally at the Water Surface," Proceedings, 12th Congress of International Association for Hydraulic Research, Vol. 4, June, 1967, pp. 47-59.

12 Harleman, D.R.F., "Innovations in Heat Disposal in the Oceans, 2nd Annual Sea Grant Lectures and Symposium (October, 1973), MIT Sea Grant Program Report, MIT, SG., 74, 7, 19 pages.

13. Iwasa, Y. and Yatsuzuka, M., "Spread of Heated Water from Vertical Multi-Port Diffuser," Proceedings, U.S.-Japan Joint Seminar on Engineering and Environmental Aspects of Heat Disposal, April, 1974.

14. Koh, R.C.Y., Brooks, N.H., List, E.J. and Wolanski, J.E., "Thermal Outfall Diffusers for the San Onofre Nuclear Power Plant," Report No. KH-R30, W.M. Keck Laboratory, C.I.T. January, 1974.

15. Liebman, C.J. and Lynn, R.W., "The Optimal Allocation of Stream Disolved Oxygen," Water Resources Research, VOL. 2, NO. 3, 1966, pp. 581-591.

16. Maass, A., et al., "Design of Water-Resources Systems," Maass, A., Editor-in-Chief, Harvard University Press, 1962, pp. 176-177, pp. 542-604.

17. Smith, I.M., Farraday, R.V. and O'Connor, "Rayleigh-Ritz and Galerkin Finite Elements for Diffusion-Convective Problems", Water Resources Research, Vol. 9, No. 3, June, 1973, pp. 593-606.

18. Sueishi, T. and Naitoo, M., "Optimization Techniques in Water Supply and Sewerage Systems," Text of 4th Lecture Meeting on Planning in Civil Engineering, JSCE, 1971, pp. 1-47, (in Japanese).

19. Tamai, N., "Unified View of Diffusion and Dispersion in Coastline Waters," Journal of Faculty of Engineering, The University of Tokyo, Vol. XXXI, NO. 4, 1972, pp. 531-692.

20. Tamai, N., Wiegel, L.R., Tornberg, F.G., "Horizontal Surface Discharge of Warm Water Jets," Journal of the Power Division, ASCE, Vol. 95, No. PO2, October, 1969, pp. 253-276.

21. Thomann, R.V., and Sobel, M.J., "Estuarine Water Quality Management and Forecasting," Journal of the Sanitary Engineering Division, ASCE, Vol. 90, No. SA5, 1964, pp. 9-36.

22. Timoshenko, P.S. and Woinowski-Krieger, S., "28. Navier Solution for Simply Supported Rectangular Plates," Theory of Plates and Shells, 2nd edition, McGraw-Hill, Kogakusha, 1959, pp. 108-113.

23. Wada, A., "Study on Prediction Method of Simulation Analysis for Diffusion of Discharged Warm Water," Proceedings, U.S.-Japan Joint Seminar on Engineering and Environmental Aspects of Heat Disposal, April, 1974.

24. Yatsuzuka, M., "Hydraulic Research on Spread of Heated Warm Water Vertical Multi-Port Diffuser," Thesis Presented to Kyoto University, in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy, February 1975, (in Japanese).

25. Zienkiewicz, O.C., "The Finite Element Method in Engineering Science," 2nd ed., McGraw-Hill, 1971, pp. 295-314, pp. 33-47, pp. 103-128.

Notations

The following symbols are used in Chapter 3:

$[A] = [a_{np}]$ = state matrix = global stiffness matrix, $(N \times N)$ matrix;

a^e_{rs} = element (factor) of element stiffness matrix;

b = constant in governing equation;

b^g_l = constant in lth constraint;

b_n = constant in nth equilibrium equation;

$c_1 \sim c_7$ = coefficients in governing equation;

c^θ_i = decision-evaluation coefficient = cost coefficients associated with ${}_j\theta_i$;

c^ϕ_n = satate-evaluation coefficient = cost coefficient associated with ϕ_n;

$[D] = [d_{ni}]$ = decision matrix; $(N \times I)$ matrix;

D_{xk} = diffusion coefficient (D_x or D_y), L^2T^{-1};

$D.E.$ = differential equation (governing equation);

f = objective function;

$[G_\theta]$ = decision-constraint matrix, $(L_T \times I)$ matrix;

$[G_\phi]$ = state-constraint matrix, $(L_T \times N)$ matrix;

$[g^\theta] = [g^\theta_{ii}]$ = sub-decision-constraint matrix, $(I \times I)$ unit matrix;

$[g^\phi] = [g^\phi_{ln}]$ = sub-state-constraint matrix, $(L \times N)$ matrix;

g = constraint;

h = boundary condition;

I = total number of decision variables (total number of controllable loads);

$i = 1 \sim I$ = decision variable number (controllable load number);

j = nodal point number associated with ith decision variable;

K = heat transfer coefficient at water surface or decay factor of pollutant, T^{-1};

k = index of coordinates;

L_1, L_2, L_3 = area coordinates;

L = total number of regulated nodal points in water quality requirement;

L_T = total number of constraints in general FELP Method;

L_x, L_y = lengths in x and y directions of water basin, L;

$l = 1 \sim L$ = water quality requirement number;

$l = 1 \sim L_T$ = constraint number in general FELP Method;

l_{xk} = direction cosine outward normal to boundary (l_x or l_y);

m = nodal point number associated with lth water quality requirement;

m, n = component numbers in x and y directions in double Fourier series;

$[N] = [N_1 \ .. \ N_r \ .. \ N_R]$ = shape functions expressed by area coordinates;

N_n, N_r, N_s = shape functions associated with nodal points n, r and s;

N = total number of state variables (total number of nodal points);

$n, p = 1 \sim N$ = state variable number (nodal point number);

$Q^c = \theta$ = decision variable (controllable load);

$\underline{Q}^c$ = lower limit of controllable load;

$\overline{Q}^c$ = upper limit of controllable load;

${}_jQ_i^c = {}_j\theta_i$ = ith decision variable (ith controllable load);

${}_j\overline{Q}_i^c$ = upper limit of ith controllable load;

Q^u = uncontrollable load;

Q_n^u = uncontrollable load at nodal point n;

Q_{mn} = known coefficient associated with given load;

$Q(x,y)$ = given load in analytical method;

q = intensity of (heat) flux per unit length of boundary;

q_n^c, q_r^c = equivalent nodal controllable loads at nodal points n and r in each element;

q_n^u, q_r^u = equivalent nodal uncontrollable loads at nodal points n and r in each element;

R = number of nodal points in each triangular element (*3* or *6*);

r, s = $1 \sim R$ = nodal point numbers in each element;

S = boundary ($S^1 + S^2$), L or L^2;

S^1, S^2 = parts of boundary S, L or L^2;

s^2 = segment of part of boundary S^2, L or L^2;

v_k = convective velocity (v_x or v_y), LT^{-1};

W_n = weighting function;

X_k = Cartesian coordinate of boundary (x , y or z), L;

x_k = Cartesian coordinate (x, y or z), L;

Z = objective function;

κ = heat transfer coefficient or decay factor of boundary, LT^{-1};

$\underline{\Theta}$ = lower limit of decision variable;

$\overline{\Theta}$ = upper limit of decision variable;

θ = decision variable;

${}_j\theta_i$ = ith decision variable;

$\underline{\Phi}$ = lower limit of state variable;

$\overline{\Phi}$ = upper limit of state variable;

Φ_a = temperature or concentration of surrounding boundary;

Φ_b = prescribed boundary value;

${}_m\overline{\Phi}_l$ = lth water quality requirement;

ϕ = state variable (water quality in water basin);

$\{\phi\}^e$ = listing of nodal values for each element;

ϕ_n, ϕ_p = nth and pth state variables (water quality at nodal points n and p);

ϕ_r, ϕ_s = discrete nodal representations of state variables of each element;

ϕ_{mn} = unknown coefficient associated with state variable in analytical method;

Ω_s = whole domain (whole water basin), L^2 or L^3;

Ω_s^g = subdomain associated with constraints, L^2 or L^3;

Δ = area of triangular element, L^2;

Notes. Area Coordinates and Shape Functions

(1). Area Coordinates

A convenient set of coordinates (area coordinates) L_1, L_2 and L_3 for triangle (see Figs. 3-2(a) and 3-2(b)) is defined by the following linear relation between these and the Cartesian coordinates:

$$x = L_1x_1 + L_2x_2 + L_3x_3 \tag{A-1}$$

$$y = L_1y_1 + L_2y_2 + L_3y_3 \tag{A-2}$$

$$1 = L_1 + L_2 + L_3 \tag{A-3}$$

To every set, L_1, L_2 and L_3 (which are not independent, but are related by Eq. A-3) corresponds a unique set of Cartesian coordinates. At point 1, $L_1 = 1$ and $L_2 = L_3 = 0$, etc. A linear relation between the new and Cartesian coordinates implies that countours of L_1 are equally placed straight lines parallel to side 2-3 on which $L_1 = 0$, etc.

Indeed it is easy to see that an alternative definition of the coordinate L_1 of a point P is given by a ratio of the shaded triangle to that of the total triangle.

$$L_1 = \frac{Area\ P23}{Area\ 123} \tag{A-4}$$

Hence the name of area coordinates is derived.

Solving Eqs. A-1, A-2 and A-3 yields the following equations.

$$L_1 = (a_1 + b_1x + c_1y)/2\Delta \tag{A-5}$$

$$L_2 = (a_2 + b_2x + c_2y)/2\Delta \tag{A-6}$$

$$L_3 = (a_3 + b_3x + c_3y)/2\Delta \tag{A-7}$$

in which

$$\Delta = \frac{1}{2} \det \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} = Area\ 123 \tag{A-8}$$

and

$$a_1 = x_2 y_3 - x_3 y_2 \tag{A-9}$$

$$b_1 = y_2 - y_3 \tag{A-10}$$

$$c_1 = x_3 - x_2 \tag{A-11}$$

with the other coefficients obtained by a cyclic permutation of subscripts in the order, *1, 2, 3.*

When element matrices have to be evaluated it will follow that we are often faced with integration of quantities defined in terms of area coordinates over triangular region. It is useful to note in this context the following integration expression

$$\int_\Delta L_1^a L_2^b L_3^c \, d\Delta = \frac{a!\, b!\, c!}{(a + b + c + 2)!} 2\Delta \tag{A-12}$$

(2). Shape Functions

In triangular element with three nodal points

In the triangular element with three nodal points (see Fig. 3-2(a)), the shape functions $[N] = [N_1 \ \ N_2 \ \ N_3]$ are simply the area coordinates.

$$N_1 = L_1, \quad N_2 = L_2, \quad N_3 = L_3 \tag{A-13}$$

In triangular element with six nodal points

In the triangular element with six nodal points (see Fig. 3-2(b)), the shape functions $[N] = [N_1 \quad N_2 \quad N_3 \quad N_4 \quad N_5 \quad N_6]$ are as follows:

$$N_1 = (2L_1 - 1)L_1, \text{ etc.} \qquad \textit{(for corner nodal points)} \tag{A-14}$$

$$N_4 = 4L_2L_3, \text{ etc.} \qquad \textit{(for mid-side nodal points)} \tag{A-15}$$

(3). Differentiation of Shape Functions

The differentiations of shape functions are as follows:

In triangular element with three nodal points

$$\frac{\partial N_1}{\partial x} = \frac{\partial L_1}{\partial x} = \frac{\partial}{\partial x}(a_1 + b_1x + c_1y)/2\Delta = \frac{b_1}{2\Delta}, \text{ etc.} \tag{A-16}$$

$$\frac{\partial N_1}{\partial y} = \frac{\partial L_1}{\partial y} = \frac{\partial}{\partial y}(a_1 + b_1x + c_1y)/2\Delta = \frac{c_1}{2\Delta}, \text{ etc.} \tag{A-17}$$

In triangular element with six nodal points

$$\frac{\partial N_1}{\partial x} = \frac{\partial(2L_1^2 - L_1)}{\partial x} = (4L_1 - 1)\frac{\partial L_1}{\partial x} = (4L_1 - 1)\frac{b_1}{2\Delta}, \text{ etc.} \tag{A-18}$$

$$\frac{\partial N_1}{\partial y} = \frac{\partial(2L_1^2 - L_1)}{\partial y} = (4L_1 - 1)\frac{\partial L_1}{\partial y} = (4L_1 - 1)\frac{c_1}{2\Delta}, \text{ etc.} \tag{A-19}$$

(for corner nodal points)

$$\frac{\partial N_4}{\partial x} = \frac{\partial(4L_2L_3)}{\partial x} = 4\left(\frac{\partial L_2}{\partial x}L_3 + L_2\frac{\partial L_3}{\partial x}\right) = \frac{2}{\Delta}(b_2L_3 + L_2b_3), \text{ etc.} \tag{A-20}$$

$$\frac{\partial N_4}{\partial y} = \frac{\partial(4L_2L_3)}{\partial y} = 4\left(\frac{\partial L_2}{\partial y}L_3 + L_2\frac{\partial L_3}{\partial y}\right) = \frac{2}{\Delta}(c_2L_3 + L_2c_3), \text{ etc.} \tag{A-21}$$

(for mid-side nodal points)

Chapter 4

EFFICIENT COMPUTATIONAL ALGORITHM FOR FINITE ELEMENT & LINEAR PROGRAMMING METHOD AND FINITE DIFFERENCE & LINEAR PROGRAMMING METHOD

Summary

An efficient computational algorithm for finite element & linear programming method (FELP Method, or, the F.E. & L.P Method) and finite difference & linear programming method (FDLP Method, or, the F.D. & L.P. Method) is studied in order to control water pollution problems. FELP Method has been developed by the combined use of finite element method with linear programming in order to solve systems of differential equations with both equality or inequality constraints and an objective function. (See Chapter 3). In a manner similar to FELP Method, FDLP Method has also been developed and systematized by using finite difference method with linear programming. (See Chapter 2). The problems formulated by FELP Method or FDLP Method could be solved by using existing computer codes for linear programming based on the two phases of the simplex method. However, in order to save computer time and memory, an efficient computational algorithm of FELP Method and FDLP Method is developed by taking note of the fact that the problems formulated by these methods have special structures. In the proposed algorithm, by using Gaussian Elimination, an initial basic feasible solution of the simplex method is obtained without the introduction of artificial variables.

4-1. General Concepts

An efficient computational algorithm for finite element & linear programming method and finite difference & linear programmng method is developed in order to control water pollution problems governed by the diffusion convection equation. Finite element & linear programming method (FELP Method, or, the F.E. & L.P. Method) has been developed by the combined use of finite element method with linear programming in order to solve systems of differential equations with both equality or inequality constraints and an objective function. (See Chapter 3). In a manner similar to FELP Method, finite difference & linear programming method (FDLP Method, or, the F.D. & L.P. Method) has also been developed and systematized by using finite difference method with linear programming. (See Chapter 2).

Generally, most practical applications of linear programming (1, 3, 4, 5 and 6) make use of degital computer and existing computer codes. The problems formulated by FELP Method or FDLP Method could be solved by using existing computer codes for linear programming based on the two phases of the simplex method. The two phases of the simplex method, however, need large computer time and memory to obtain an initial basic feasible solution. In this chapter, in order to save computer time and memory, an efficient computational algorithm for FELP Method and FDLP Method is studied by taking note of the fact that the problems formulated by these methods have special structures. By using Gaussian Elimination, an initial basic feasible solution of the simplex method is obtained without the introduction of artificial variables. The details of the algorithm are shown through a numerical example of FELP Method.

4-2. Initial Basic Feasible Solution

4-2-1. Conversion of Linear Inequality Systems to Standard Systems

The obtained matrix-vector forms of FELP Method and FDLP Method in water pollution control are as follows (see Eqs. 17-21 in Chapter 3 and Chapter 2):

Equilibrium Equations (N-Eqs.)

$$[\,A\,]\{\phi_n\} + [\,D\,]\{{}_jQ_i^c\} = +\{Q_n^u\} \quad \textit{(for FELP Method)}$$

$$[\,A\,]\{\phi_n\} + [\,D\,]\{{}_jQ_i^c\} = -\{Q_n^u\} \quad \textit{(for FDLP Method)} \qquad (17) \qquad (1)$$

Constraints $((L_T = L+I)\text{-Eqs.})$

$$[\,g^{\phi}\,]\{\phi_n\} \leqq \{{}_m\overline{\Phi}_l\} \qquad (18) \qquad (2)$$

$$[\,g^{\theta}\,]\{{}_jQ_i^c\} \leqq \{{}_j\overline{Q}_i^c\} \qquad (19) \qquad (3)$$

Nonnegative Conditions

$$\phi_n \geqq 0 \ (n = 1 \sim N), \quad {}_jQ_i^c \geqq 0 \ (i = 1 \sim I) \qquad (20) \qquad (4)$$

Objective Function

$$Z = \underset{\{{}_jQ_i^c\}}{\text{Opt.}}\ f(\{\phi_n\}, \{{}_jQ_i^c\}) = \underset{\{{}_jQ_i^c\}}{\text{Opt.}} \left(\sum_{n=1}^{N} c_n^{\phi}\,\phi_n + \sum_{i=1}^{I} c_i^{\theta}\,{}_jQ_i^c \right)$$

$$\simeq \underset{\{{}_jQ_i^c\}}{\text{Max.}} \sum_{i=1}^{I} c_i^{\theta}\,{}_jQ_i^c \quad \begin{array}{l}(c_i^{\theta} = 1 \ \textit{for FELP Method}) \\ (c_i^{\theta} = {}_ja_i \ \textit{for FDLP Method})\end{array} \qquad (21) \qquad (5)$$

in which $[A] = [a_{np}]$ = the state matrix (the global stiffness matrix), $(N\times N)$ matrix; $[D] = [d_{ni}]$ = the decision matrix, $(N\times I)$ matrix, $[g^{\phi}]$ =

$= [g^{\phi}_{ln}]$ = the sub-state-constraint matrix, $(L \times N)$ matrix; $[g^{\theta}] = [g^{\theta}_{ii}]$ = the sub-decision-constraint matrix, $(I \times I)$ unit matrix; ϕ_n = nth state variable = water quality at the nodal (mesh) point n; ${}_jQ^c_i = {}_j\theta_i$ = ith decision variable = ith controllable load; j = nodal (mesh) point number associated with ith decision variable; Q^u_n = constant in nth equilibrium equation = uncontrollable load at the nodal (mesh) point n; ${}_j\overline{Q}^c_i$ = the upper limit of ith controllable load; ${}_m\overline{\Phi}_l$ = lth water quality requirement; m = regulated nodal (mesh) point number in lth water quality requirement; c^{ϕ}_n = state-evaluation coefficient = cost coefficient associated with ϕ_n; c^{θ}_i = decision-evaluation coefficient = cost coefficient associated with ${}_jQ^c_i$; $i = 1 \sim I$ (I: total number of the decision variables, i.e., total number of the controllable loads); $n = 1 \sim N$ (N: total number of the state variables, i.e., total number of the nodal (mesh) points); $l = 1 \sim L$ (L: total number of the regulated nodal (mesh) points in water quality requirements); $L_T = L + I$ = total number of the constraints.

In the maximization problem of total of the controllable loads, all of the cost coefficients $\{c^{\phi}_n\}$ are equal to zero, and all of the cost coefficients $\{c^{\theta}_i\}$ are equal to '1' (for FELP Method) or '${}_ja_i$ = area governed by mesh point j' (for FDLP Method) as shown in Eq. 5.

In the followings, in order to avoid the combersome in the symborical treatments, the following symboles are used in stead of ${}_jQ^c_i$, $\pm Q^u_n$, ${}_m\overline{\Phi}_l$ and ${}_j\overline{Q}^c_i$.

$$\theta_i = {}_jQ^c_i, \quad b_n = \pm Q^u_n, \quad b^{\phi}_l = {}_m\overline{\Phi}_l, \quad b^{\theta}_i = {}_j\overline{Q}^c_i \tag{6}$$

The discretized equation systems (Eqs. 1-5) contain inequality constraints (Eqs. 2 and 3). In developing solution procedures of FELP Method or FDLP Method, we shall find much easier to work with equality constraints. By converting the inequalities to the equalities through the use of nonnegative variables, or slack variables $\{\mu_l\}$ and $\{\nu_i\}$, we obtain the standard systems (1) expressed in linear simultaneous equations.

Equilibrium Equations (N-Eqs.)

$$[\ A\ ]\{\phi_n\} + [\ D\ ]\{\theta_i\} = \{b_n\} \tag{7}$$

Constraints (($L_T = L+I$*)-Eqs.)*

$$[\ g^{\phi}\ ]\{\phi_n\} + \{\mu_l\} = \{b_l^{\phi}\} \tag{8}$$

$$[\ g^{\theta}\ ]\{\theta_i\} + \{\nu_i\} = \{b_i^{\theta}\} \tag{9}$$

Nonnegative Conditions

$$\phi_n \geqq 0 \ (n = 1 \sim N), \quad \theta_i \geqq 0 \ (i = 1 \sim I) \tag{10-1}$$

$$\mu_l \geqq 0 \ (l = 1 \sim L), \quad \nu_i \geqq 0 \ (i = 1 \sim I) \tag{10-2}$$

Objective Function

$$Z = \underset{\{\theta_i\}}{Opt.}\ f\,(\{\phi_n\},\ \{\theta_i\}) = \underset{\{\theta_i\}}{Opt.}\left(\sum_{n=1}^{N} c_n^{\phi}\,\phi_n + \sum_{i=1}^{I} c_i^{\theta}\,\theta_i\right)$$

$$\simeq \underset{\{\theta_i\}}{Max.} \sum_{i=1}^{I} c_i^{\theta}\,\theta_i \tag{11}$$

In the above standard systems, the number of the linear simultaneous equations is $(N+L_T)$ and the number of the variables is $(N+I+L_T)$.

4-2-2. Initial Basic Feasible Solution and Canonical Systems

The obtained standard systems are solved by using the simplex method (1, 3, 4). Before the use of the simplex method, however, it is necessary to find a solution which satisfies Eqs. 7-10. Such a solution which satisfies the given conditions (Eq. 7-10) is called basic feasible solution. The simplex method is always initiated with a program whose equations are in canonical systems which apparently show a basic feasible solution (1). In the case where a canonical systems (1), or an initial basic feasible solution, can not be found, we have to use the two phases of the simplex method. The first phase of the two phases of the simplex method augments the systems to include a basic set of artificial variables $\{\chi_n\}$, $\{\chi_l^\phi\}$ and $\{\chi_i^\theta\}$ as follows:

Equilibrium Equations (N-Eqs.)

$$[\,A\,]\{\phi_n\} + [\,D\,]\{\theta_i\} \qquad + \{\chi_n\} \qquad = \{b_n\} \tag{12}$$

Constraints ((L_T = L+I)-Eqs.)

$$[\,g^\phi\,]\{\phi_n\} \qquad + \{\mu_l\} \qquad + \{\chi_l^\phi\} \qquad = \{b_l^\phi\} \tag{13}$$

$$[\,g^\theta\,]\{\theta_i\} \qquad + \{\nu_i\} \qquad + \{\chi_i^\theta\} = \{b_i^\theta\} \tag{14}$$

Nonnegative Conditions

$$\phi_n \geqq 0 \ (n = 1 \sim N), \quad \theta_i \geqq 0 \ (i = 1 \sim I) \tag{15-1}$$

$$\mu_l \geqq 0 \ (l = 1 \sim L), \quad \nu_i \geqq 0 \ (i = 1 \sim I) \tag{15-2}$$

$$\chi_n \geqq 0 \ (n = 1 \sim N), \ \chi_l^\phi \geqq 0 \ (l = 1 \sim L), \ \chi_i^\theta \geqq 0 \ (i = 1 \sim I) \tag{15-3}$$

Objective Function

$$Z = Min. \left(\sum_{n=1}^{N} \chi_n + \sum_{l=1}^{L} \chi_l^{\phi} + \sum_{i=1}^{I} \chi_i^{\theta} \right) \tag{16}$$

In the above canonical systems the number of the simultaneous equations is $(N+L_T)$ and the number of the variables is $(N+I+L_T+N+L_T)$. Therefore, the first phase of the two phases of the simplex method requires considerable computer time and memory because of the introduction of the artificial variables.

In this research, however, by taking note of the fact that FELP Method and FDLP Method in the present problem have special structures, an efficient computational algorithm to obtain an initial canonical systems, or an initial basic feasible solution, is developed without the introduction of artificial variables. In the followings the details of the algorithm are mentioned.

At first the equilibrium equations (Eq. 7) in the standard systems are transformed by using Gaussian Elimination (2) as shown below. (As for the details, see Notes).

$$[\,\mathrm{AI}\,]\{\phi_n\} + [\,\mathrm{D}'\,]\{\theta_i\} = \{B_n\} \tag{17}$$

, or

$$\{\phi_n\} = \{B_n\} - [\,\mathrm{D}'\,]\{\theta_i\} \tag{18}$$

in which $[\mathrm{AI}]$ = the transformed state matrix, $(N{\times}N)$ unit matrix; $[\mathrm{D}']$ = the transformed decision matrix, $(N{\times}I)$ matrix; and $\{B_n\}$ = transformed constant vector.

The transformed equilibrium equations mean solution for the state variables $\{\phi_n\}$ expressed in the decision variables $\{\theta_i\}$.

Substitution of Eq. 18 into Eq. 8 yields the following equation.

$$[\ W\ ]\{\theta_i\} + \{\mu_l\} = \{B_l^{\phi}\} \tag{19}$$

where

$$[\ W\ ] = -\ [\ g^{\phi}\ ][\ D'\ ] \tag{20}$$

$$\{B_l^{\phi}\} = \{b_l^{\phi}\} - [\ g^{\phi}\ ]\{B_n\} \tag{21}$$

Replacing Eqs. 7 and 8 by Eqs. 17 and 19, we obtain the initial canonical systems in the efficient computational algorithm of FELP Method and FDLP Method, that is:

Equilibrium Equations (N-Eqs.)

$$[AI\ ]\{\phi_n\} + [\ D'\ ]\{\theta_i\} = \{B_n\} \tag{22}$$

Constraints ($(L_T = L+I)$-Eqs.)

$$[\ W\ ]\{\theta_i\} + \{\mu_l\} = \{B_l^{\phi}\} \tag{23}$$

$$[\ g^{\theta}\ ]\{\theta_i\} + \{\nu_i\} = \{b_i^{\theta}\} \tag{24}$$

Nonnegative Conditions

$$\phi_n \geqq 0\ (n = 1 \sim N), \quad \theta_i \geqq 0\ (i = 1 \sim I) \tag{25-1}$$

$$\mu_l \geqq 0\ (l = 1 \sim L), \quad \nu_i \geqq 0\ (i = 1 \sim I) \tag{25-2}$$

Objective Function

$$Z = \underset{\{\theta_i\}}{Opt.}\ f\ (\{\phi_n\}, \{\theta_i\}) = \underset{\{\theta_i\}}{Opt.}\ \left(\sum_{n=1}^{N} c_n^{\phi}\ \phi_n + \sum_{i=1}^{I} c_i^{\theta}\ \theta_i \right)$$

$$\simeq \underset{\{\theta_i\}}{Max.} \sum_{i=1}^{I} c_i^{\theta}\ \theta_i \tag{26}$$

Since [AI] and $[g^{\theta}]$ are unit matrices, we can easily find a solution which satisfies the given conditions (Eqs. 22-25), that is:

$$\{\phi_n\} = \{B_n\} \quad (n = 1 \sim N) \tag{27-1}$$

$$\{\theta_i\} = \{0\} \quad (i = 1 \sim I) \tag{27-2}$$

$$\{\mu_l\} = \{B_l^{\phi}\} \quad (l = 1 \sim L) \tag{27-3}$$

$$\{\nu_i\} = \{b_i^{\theta}\} \quad (i = 1 \sim I) \tag{27-4}$$

This is the initial basic feasible solution in the efficient computational algorithm of FELP Method and FDLP Method. Therefore, in the initial canonical systems, the basic variables are $\{\phi_n\}$, $\{\mu_l\}$ and $\{\nu_i\}$, and the non-basic variables are $\{\theta_i\}$, respectively.

Since in the initial basic feasible solution all of the decision variables $\{\theta_i\}$, or all of the controllable loads $\{{}_jQ_i^c\}$, are equal to zero, the initial basic feasible solution for the state variables, $\{B_n\}$, gives us the distribution pattern of the water quality which arises exclusively from the uncontrollable loads $\{Q_n^u\}$.

4-3. Improvement of Basic Feasible Solution

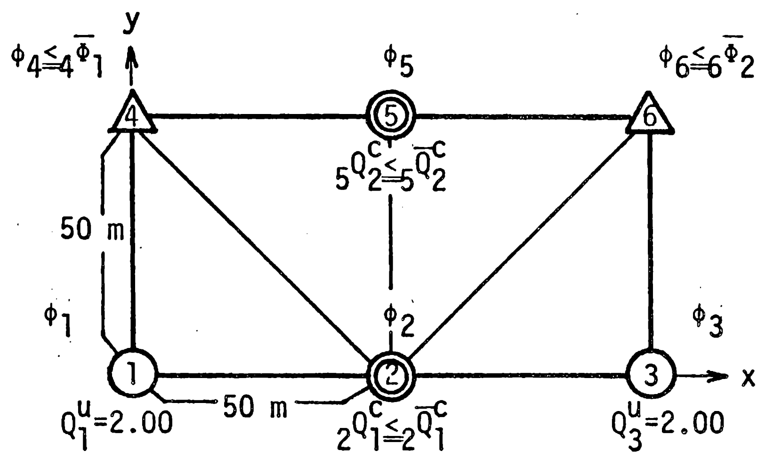
4-3-1. Formulation of FELP Method in a Simple Model Basin

In order to improve the basic feasible solution, the simplex method is used for the canonical systems (Eqs. 22-26). As for the details of the simplex method, one may follow Dantzig (1) or Gass (3). The optimization procedures (the maximization procedures of total of the decision variables in the present instance) are explained through a numerical example of FELP Method in a rectangular model basin with non-conductive boundaries as shown in Fig. 4-1. The obtained matix-vector forms of FELP Method are as follows:

$$
\begin{array}{cc}
 & \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 1 & 2 \end{array} \\
\begin{array}{c} 1\\2\\3\\4\\5\\6\\1\\2\\3\\4 \end{array} &
\left[\begin{array}{cccccc|cc}
1.21 & -0.40 & 0.00 & -0.40 & 0.00 & 0.00 & 0 & 0 \\
-0.40 & 2.83 & -0.40 & 0.21 & -0.80 & 0.21 & -1 & 0 \\
0.00 & -0.40 & 1.21 & 0.00 & 0.00 & -0.40 & 0 & 0 \\
-0.40 & 0.21 & 0.00 & 1.42 & -0.40 & 0.00 & 0 & 0 \\
0.00 & -0.80 & 0.00 & -0.40 & 2.42 & -0.40 & 0 & -1 \\
0.00 & 0.21 & -0.40 & 0.00 & -0.40 & 1.42 & 0 & 0 \\
\hline
0 & 0 & 0 & 1 & 0 & 0 & \multicolumn{2}{c}{\text{Zero}} \\
0 & 0 & 0 & 0 & 0 & 1 & \multicolumn{2}{c}{\text{Matrix}} \\
\hline
\multicolumn{6}{c|}{} & 1 & 0 \\
\multicolumn{6}{c|}{\text{Zero Matrix}} & 0 & 1
\end{array}\right]
\end{array}
\begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \theta_1 \\ \theta_2 \end{Bmatrix}
\begin{array}{ll}
= b_1 & (= 2.00) \\
= b_2 & (= 0) \\
= b_3 & (= 2.00) \\
= b_4 & (= 0) \\
= b_5 & (= 0) \\
= b_6 & (= 0) \\
\leqq b_1^\phi & (= 3.00) \\
\leqq b_2^\phi & (= 3.00) \\
\leqq b_1^\theta & (= 5.00) \\
\leqq b_2^\theta & (=20.00)
\end{array}
\begin{array}{c} \\ \\ (28) \\ \\ \\ \\ (29) \\ \\ (30) \\ \\ \end{array}
$$

$$\phi_n \geqq 0 \ (n = 1 \sim 6), \qquad \theta_i \geqq 0 \ (i = 1 \sim 2) \tag{31}$$

$$Z = \underset{\{\theta_i\}}{Opt.} \left(\sum_{n=1}^{6} c_n^\phi \phi_n + \sum_{i=1}^{2} c_i^\theta \theta_i \right) = \underset{\{\theta_i\}}{Max.} (\theta_1 + \theta_2) \tag{32}$$



D_x = 1.0 m^2/sec

D_y = 1.0 m^2/sec

K = 0.001 ^{1}/sec

$v_x = v_y = 0$

N = 6, I = 2

L = 2

L_T = L + I
= 2 + 2 = 4

$\partial\phi/\partial n = 0$ (On the four boundaries)

${}_2\bar{Q}_1^c$ = 5.00

${}_5\bar{Q}_2^c$ = 20.00

${}_4\bar{\Phi}_1 = {}_6\bar{\Phi}_2$ = 3.00

◎: Nodal point for controllable load

○: Nodal point for uncontrollable load

△: Regulated nodal point in water quality requirement

Fig. 4-1. Input Data on FELP Method in a Simple Model Basin

4-3-2. Improving Solution by Simplified Simplex Method

The transformation of Eqs. 28-32 by using Eqs. 22-26 yields the initial simplex tableau as shown in Table 4-1(a). Table 4-1(a) is rewritten in the general form as shown in Table 4-1(a)´. The transformed decision-constraint matrix $[G'_\theta]$ in these tables is as follows:

$$\underset{(L_T\times(I+L_T))}{[\ R\ ]} = \begin{bmatrix} G'_\theta & R' \\ & R^2 \end{bmatrix} \quad (33\text{-}1) \qquad \underset{(L_T\times I)}{[\ G'_\theta\ ]} = \begin{bmatrix} W \\ \\ g^\theta \end{bmatrix} = \begin{bmatrix} \lambda^0_{11} & \cdots & \lambda^0_{1i} & \cdots & \lambda^0_{1I} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \lambda^0_{s1} & \cdots & \lambda^0_{si} & \cdots & \lambda^0_{sI} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \lambda^0_{L_T1} & \cdots & \lambda^0_{L_Ti} & \cdots & \lambda^0_{L_TI} \end{bmatrix} \quad (33\text{-}2)$$

In Tables 4-1(a) or 4-1(a)´ we can find that the simplex tableau for the maximization of total of the decision variables has the following specialities.

1). All of the cost coefficient associated with state variables and slack variables are equal to zero.

2). All of the elements of the transformed decision constraint matrix $[D']$ are negative.

3). Optimarity criteria are negative only for the decision variables.

4). All of the state variables are always the basic variables.

5). Replacements of the basic variables by non-basic variables are carried out between the slack variables and the decision variables. (This means the pivot element is selected within $[\ R\ ]$ matrix).

Therefore, the simplex procedures in the present problem are simplified in each cycle (cycle k) as follows:

1). The testing of the optimal criteria (simplex criteria) associated with the decision variables to determine whether a maximum solution has been found, i.e., whether $\zeta_i \geqq 0$ for all i.

The optimality criteria is

$$\zeta_i = z_i - c_i^{\theta} = z_i - 1 = \sum_{s=1}^{L_T} c_s^k \lambda_{si}^k - 1 \quad (i = 1 \sim I) \tag{34-1}$$

$$\zeta_i = z_i - 0 = \sum_{s=1}^{L_T} c_s^k \lambda_{si}^k \qquad (i = I + 1 \sim I + L_T) \tag{34-2}$$

2). The selection of the non-basic variable to be introduced into the basis (the set of the basic variables) with

$$i^* \Leftarrow \underset{i}{Max.}\ [Absolute(\zeta_i)] \quad (random\ selection\ for\ ties) \tag{35}$$

3). The selection of the slack variable to be eliminated from the basis with

$$s^p \Leftarrow \xi_{s^p} = \underset{s}{Min.}\ [\beta_s^k/\lambda_{si^*}^k],\ \lambda_{si^*}^k > 0 \quad (random\ selection\ for\ ties) \tag{36}$$

4). The transformation of the tableau with the followoing pivot operation (the pivot element is $\lambda_{s^p i^*}^k$)

$$B_n^{k+1} = B_n^k - \delta_{ni^*}^k \xi_{s^p} \quad (n = 1 \sim N) \tag{37}$$

$$\beta_s^{k+1} = \beta_s^k - \lambda_{si^*}^k \xi_{s^p} \quad (s = 1 \sim s^p - 1\ \&\ s^p + 1 \sim L_T) \tag{38-1}$$

$$\beta_{s^p}^{k+1} = \xi_{s^p} \qquad (s = s^p) \tag{38-2}$$

$$\delta_{ni}^{k+1} = \delta_{ni}^k - \delta_{ni^*}^k \lambda_{s^p i}^k/\lambda_{s^p i^*}^k \quad (n = 1 \sim N,\ i = 1 \sim I + L_T) \tag{39}$$

$$\lambda_{si}^{k+1} = \lambda_{si}^k - \lambda_{si^*}^k \lambda_{s^p i}^k/\lambda_{s^p i^*}^k \quad (s = 1 \sim s^p - 1\ \&\ s^p + 1 \sim L_T,\ i = 1 \sim I + L_T) \tag{40-1}$$

$$\lambda_{s^p i}^{k+1} = \lambda_{s^p i}^k/\lambda_{s^p i^*}^k \quad (s = s^p,\ i = 1 \sim i^* - 1\ \&\ i^* + 1 \sim I + L_T) \tag{40-2}$$

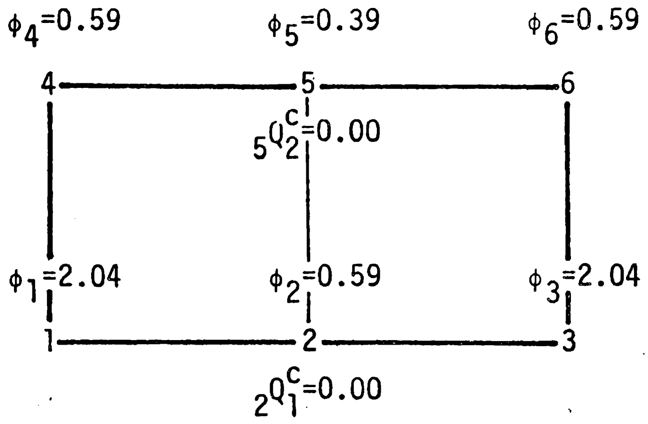
$$\lambda_{s^p i^*}^{k+1} = 1 \qquad (s = s^p,\ i = i^*) \tag{40-3}$$

	Basic variables	(1) Cost coefficients	(2) Constants	Cost coefficients: 0	0	0	0	0	0	1	1	0	0	0	0
				ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	θ_1	θ_2	μ_1	μ_2	ν_1	ν_2
	ϕ_1	0	2.04	1	0	0	0	-0	0	-0.15	-0.10	0	0	0	0
	ϕ_2	0	0.59	0	1	0	0	0	0	-0.43	-0.15	0	0	0	0
	ϕ_3	0	2.04	0	0	1	0	-0	0	-0.15	-0.10	0	0	0	0
	ϕ_4	0	0.59	0	0	0	1	0	0	-0.02	-0.15	0	0	0	0
	ϕ_5	0	0.39	0	0	0	0	1	0	-0.15	-0.51	0	0	0	0
s	ϕ_6	0	0.59	0	0	0	0	0	1	-0.02	-0.15	0	0	0	0
1	μ_1	$c_1^0=0$	2.41	0	0	0	0	0	0	0.02	0.15	1	0	0	0
2	μ_2	$c_2^0=0$	2.41	0	0	0	0	0	0	0.02	0.15	0	1	0	0
3	ν_1	$c_3^0=0$	5.00	0	0	0	0	0	0	$\boxed{1.00}$	0.00	0	0	1	0
$L_T=4$	ν_2	$c_4^0=0$	20.00	0	0	0	0	0	0	0.00	1.00	0	0	0	1
	$Z = \Sigma\,((1)\times(2))$			0	0	0	0	0	0	$\zeta_1=-1$	$\zeta_2=-1$	$\zeta_3=0$	$\zeta_4=0$	$\zeta_5=0$	$\zeta_6=0$
	$= 0.00$								i	1	2	3	4	5	$I+L_T=6$
				Optimality criteria											

	Basic variables	(1) Cost coefficients	(2) Constants	Cost coefficients											
				$c_1^\phi=0$	$c_2^\phi=0$	$c_3^\phi=0$	$c_4^\phi=0$	$c_5^\phi=0$	$c_6^\phi=0$	$c_1^\theta=1$	$c_2^\theta=1$	0	0	0	0
				ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	θ_1	θ_2	μ_1	μ_2	ν_1	ν_2
	ϕ_1	$c_1^\phi=0$	$B_1^0=B_1$	1	0	0	0	0	0	δ_{11}^0	δ_{12}^0	δ_{13}^0	δ_{14}^0	δ_{15}^0	δ_{16}^0
	ϕ_2	$c_2^\phi=0$	$B_2^0=B_2$	0	1	0	0	0	0	δ_{21}^0	δ_{22}^0	δ_{23}^0	δ_{24}^0	δ_{25}^0	δ_{26}^0
	ϕ_3	$c_3^\phi=0$	$B_3^0=B_3$	0	0	1	0	0	0	δ_{31}^0	δ_{32}^0	δ_{33}^0	δ_{34}^0	δ_{35}^0	δ_{36}^0
	ϕ_4	$c_4^\phi=0$	$B_4^0=B_4$	0	0	0	1	0	0	δ_{41}^0	δ_{42}^0	δ_{43}^0	δ_{44}^0	δ_{45}^0	δ_{46}^0
	ϕ_5	$c_5^\phi=0$	$B_5^0=B_5$	0	0	0	0	1	0	δ_{51}^0	δ_{52}^0	δ_{53}^0	δ_{54}^0	δ_{55}^0	δ_{56}^0
s	ϕ_6	$c_6^\phi=0$	$B_6^0=B_6$	0	0	0	0	0	1	δ_{61}^0	δ_{62}^0	δ_{63}^0	δ_{64}^0	δ_{65}^0	δ_{66}^0
1	μ_1	$c_1^0=0$	$\beta_1^0=B_1^\phi$	0	0	0	0	0	0	λ_{11}^0	λ_{12}^0	λ_{13}^0	λ_{14}^0	λ_{15}^0	λ_{16}^0
2	μ_2	$c_2^0=0$	$\beta_2^0=B_2^\phi$	0	0	0	0	0	0	λ_{21}^0	λ_{22}^0	λ_{23}^0	λ_{24}^0	λ_{25}^0	λ_{26}^0
3	ν_1	$c_3^0=0$	$\beta_3^0=b_1^\theta$	0	0	0	0	0	0	λ_{31}^0	λ_{32}^0	λ_{33}^0	λ_{34}^0	λ_{35}^0	λ_{36}^0
$L_T=4$	ν_2	$c_4^0=0$	$\beta_4^0=b_2^\theta$	0	0	0	0	0	0	λ_{41}^0	λ_{42}^0	λ_{43}^0	λ_{44}^0	λ_{45}^0	λ_{46}^0
	$Z = \Sigma\,((1)\times(2))$			0	0	0	0	0	0	$\zeta_1=-1$	$\zeta_2=-1$	$\zeta_3=0$	$\zeta_4=0$	$\zeta_5=0$	$\zeta_6=0$
	$= 0.00$								i	1	2	3	4	5	$I+L_T=6$
				Optimality criteria											

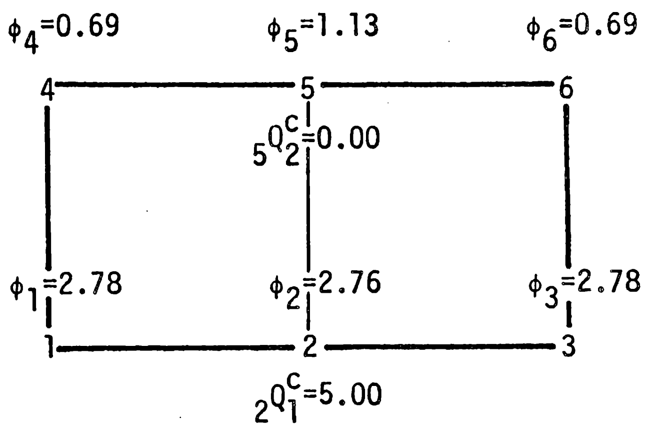
	Basic variables	(1) Cost coefficients	(2) Constants	Cost coefficients											
				0	0	0	0	0	0	1	1	0	0	0	0
				ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	θ_1	θ_2	μ_1	μ_2	ν_1	ν_2
	ϕ_1	0	2.78	1	0	0	0	0	0	0	-0.10	0	0	0.15	0
	ϕ_2	0	2.76	0	1	0	0	0	0	0	-0.15	0	0	0.43	0
	ϕ_3	0	2.78	0	0	1 AI	0	0	0	0	-0.10	0	0	0.15	0
	ϕ_4	0	0.69	0	0	0	1	0	0	0	D´ -0.15	0	0	0.02	0
	ϕ_5	0	1.13	0	0	0	0	1	0	0	-0.51	0	0	0.15	0
s	ϕ_6	0	0.69	0	0	0	0	0	1	0	-0.15	0	0	0.02	0
1	μ_1	$c_1^1=0$	2.31	0	0	0	0	0	0	0	0.15	1	0	R´ -0.02	0
2	μ_2	$c_2^1=0$	2.31	0	0	0	0	0	0	0	$\boxed{0.15}$ R	0	1	-0.02	0
3	θ_1	$c_3^1=1$	5.00	0	0	0	0	0	0	1	$G_\theta´$ 0.00	0	0	R´´ 1.00	0
$L_T=4$	ν_2	$c_4^1=0$	20.00	0	0	0	0	0	0	0	1.00	0	0	0.00	1
	$Z = \Sigma\,((1)\times(2))$			0	0	0	0	0	0	$\zeta_1=0$	$\zeta_2=-1$	$\zeta_3=0$	$\zeta_4=0$	$\zeta_5=1.00$	$\zeta_6=0$
	$= 5.00$			Optimality criteria i						1	2	3	4	5	$I+L_T=\iota$

s	Basic variable	(1) Cost coefficients	(2) Constants	Cost coefficients											
				0	0	0	0	0	0	1	1	0	0	0	0
				ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	θ_1	θ_2	μ_1	μ_2	ν_1	ν_2
	ϕ_1	0	4.30	1	0	0	0	0	0	0	0	0	0.66	0.14	0
	ϕ_2	0	5.07	0	1	0	0	0	0	0	0	0	1.00	0.41	0
	ϕ_3	0	4.30	0	0	1	0	0	0	0	0	0	0.66	0.14	0
							AI				D'				
	ϕ_4	0	3.00	0	0	0	1	0	0	0	0	0	1.00	0.00	0
	ϕ_5	0	9.11	0	0	0	0	1	0	0	0	0	3.45	0.08	0
s	ϕ_6	0	3.00	0	0	0	0	0	1	0	0	0	1.00	0.00	0
1	μ_1	$c_1^2=0$	0.00	0	0	0	0	0	0	0	0	1	-1.00	0.00	0
													R'		
2	θ_2	$c_2^2=1$	16.62	0	0	0	0	0	0	0	1	0	6.75	-0.13	0
										G'_θ		R			
3	θ_1	$c_3^2=1$	5.00	0	0	0	0	0	0	1	0	0	0.00	1.00	0
													R''		
$L_T=4$	ν_2	$c_4^2=0$	4.38	0	0	0	0	0	0	0	0	0	-6.75	0.13	1
	$Z = \Sigma((1)\times(2))$			0	0	0	0	0	0	$\zeta_1=0$	$\zeta_2=0$	$\zeta_3=0$	$\zeta_4=6.75$	$\zeta_5=0.87$	$\zeta_6=0$
	$= 20.62$								i	1	2	3	4	5	$I+L_T=6$
				Optimality criteria											



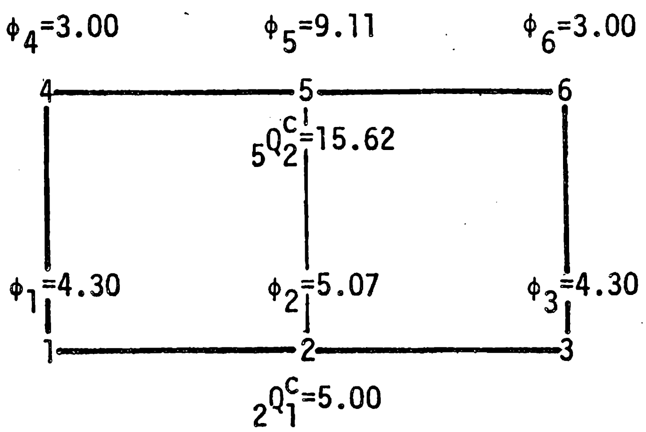
$$Z = \theta_1 + \theta_2 = {}_2Q_1^c + {}_5Q_2^c$$
$$= 0.00 + 0.00 = 0.00$$

Fig. 4-2(a). Solution in Cycle 0



$$Z = \theta_1 + \theta_2 = {}_2Q_1^c + {}_5Q_2^c$$
$$= 5.00 + 0.00 = 5.00$$

Fig. 4-2(b). Solution in Cycle 1



$$Z = \theta_1 + \theta_2 = {}_2Q_1^c + {}_5Q_2^c$$
$$= 5.00 + 15.62 = 20.62$$

Fig. 4-2(c). Solution in Cycle 2 (Optimal Solution)

The above simplified simplex procedures are applied to the numerical example as shown below.

Cycle 0 (See Table 4-1(a), (a)′ and Fig. 4-2(a)).

The initial basis consists of ϕ_1, ϕ_2, ϕ_3, ϕ_4, ϕ_5, ϕ_6, μ_1, μ_2, ν_1 and ν_2, and the solution is $(\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \mu_1, \mu_2, \nu_1, \nu_2) = (2.04, 0.59, 2.04, 0.59, 0.39, 0.59, 2.41, 2.41, 5.00, 20.00)$. Since all of the cost coefficient associated with the basis are equal to zero, the corresponding value of the objective function equals zero. In Fig. 4-2(a) the distribution pattern of the water quality arises exclusively from the uncontrollable loads $(Q_1^u = Q_3^u = 2.00)$.

In Table 4-1(a) θ_1 is selected to go into the basis, since

$$1 = i^* \Leftarrow \underset{i}{Max.}\ [Absolute\ (\zeta_i)] = -\zeta_1\ and\ -\zeta_2 = 1.00 \tag{41}$$

ξ_{s^p} is the minimum of $\beta_s^0/\lambda_{s1}^0 > 0$, that is,

$$3 = s^p \Leftarrow \xi_3 = \underset{s}{Min.}\ [2.41/0.02,\ 2.41/0.02,\ 5.00/1.00] = 5.00/1.00 \tag{42}$$

and hence ν_1 is eliminated. Therefore, the pivot element is $\lambda_{31}^0 = 1.00$. By using Eqs. 37-40 and 34, we transform the tableau and obtain a new tableau as shown in Table 4-1(b).

Cycle 1 (See Table 4-1(b) and Fig. 4-2(b)).

In Table 4-1(b) the solution is $(\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \mu_1, \mu_2, \theta_1, \nu_2) = (2.78, 2.76, 2.78, 0.69, 1.13, 0.69, 2.31, 2.31, 5.00, 20.00)$ and the value of the objective function is 5.00. In Fig. 4-2(b) the distribution pattern of the water quality arises from the controllable load $({}_2Q_1^c = 5.00)$ and the uncontrollable loads $(Q_1^u = Q_3^u = 2.00)$.

In Table 4-1(b), since the optimality criterion with negative value is only $\zeta_2 = -1.00$ for θ_2 and

$$2 = s^o = \xi_2 \text{ and } \xi_1 = \underset{s}{Min.}\ [2.31/0.15,\ 2.31/0.15,\ 20.00/1.00] = 2.31/0.15 \quad (43)$$

θ_2 is introduced into the basis and μ_2 is eliminated. Therefore, the pivot element is $\lambda^1_{22} = 0.15$. We transform the tableau and obtain a new tableau as shown in Table 4-1(c).

Cycle 2 (See Table 4-1(c) and Fig. 4-2(c)).

In Table 4-1(c) the solution is $(\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \mu_1, \theta_2, \theta_1, \nu_2) = (4.30, 5.07, 4.30, 3.00, 9.11, 3.00, 0.00, 15.62, 5.00, 4.38)$ and the value of the objective function is 20.62. Since all of the optimality criteria are nonnegative, this solution is the maximum feasible solution to be solved. In Fig. 4-2(c) the distribution pattern of the water quality arises from the optimal controllable loads $({}_2Q^c_1 = 5.00$ and ${}_5Q^c_2 = 15.62)$ and the uncontrollable loads $(Q^u_1 = Q^u_3 = 2.00)$.

Thus, the optimal solution is obtained.

4-4. Concluding Remarks

An efficient computational algorithm for finite element & linear programming method (FELP Method, or, the F.E. & L.P. Method) and finite difference & linear programming method (FDLP Method, or, the F.D. & L.P. Method) was described through the application to the control of diffusion-convection phenomena in water pollution problems. By using Gaussian Elimination, the initial basic feasible solution of FELP Method and FDLP Method was obtained without the introduction of the artificial variables. Therefore, the proposed algorithm considerably save computer time and memory. In Gaussian Elimination the use of the skillful techniques based on band matrix of finite element method will bring to further reduction of computer time and memory.

The computational work in this chapter was performed by utilizing HITAC 8700/8800 at the computer center of the University of Tokyo and the program library for the simplex method "HI/TC/LP02" (made by K. Ikura and revised by Hitachi Co. Ltd.).

References

1. Dantzig, B.G., "Linear Programming and Extensions," Princeton University Press, 1963.
2. Doherty, P.W., Wilson, L.E. and Taylor, L.R., "Stress Analysis of Axisymmetric Solids Utilizing Higher-Order Quadrilateral Finite Elements," Report No. 69-3, Structural Engineering Laboratory, University of California, Berkeley, California, January, 1969, pp. A1-A5.
3. Gass, I.S., "Linear Programming, Methods and Applications," 3rd ed., McGraw-Hill, Kogakusha, Ltd., 1969.
4. Gue, L.R. and Thomas, E.M., "Mathematical Methods in Operations Research," The Macmillan Company, pp. 273-290.
5. Koyama, A., "Introduction of Linear Programming," Nihon Keizai Shibun Sha, 1966, (in Japanese).
6. Orchard-Hays, W., "Advanced Linear-Programming Computing Techniques," McGraw-Hill, 1968.

Notations

The following symbols are used in Chapter 4:

$[A] = [a_{np}]$ = state matrix (global stiffness matrix), $(N\times N)$ matrix;

${}_ja_i$ = area governed by mesh point j;

$[AI]$ = transformed state matrix, $(N\times N)$ unit matrix;

$\{B_n\}$ = constant vector of transformed equilibrium equations;

$\{B_l^\phi\}$ = constant vector of transformed sub-constraints;

$\{B_n^k\}$ = constant vector in equilibrium equations in k cycle simplex tableau;

$\{b_n\} = \pm\,\{Q_n^u\}$ = constant vector in equilibrium equations (+: for FELP Method, -: for FDLP Method);

$b_l^\phi = {}_m\overline{\Phi}_l$

$b_i^\theta = {}_j\overline{Q}_i^c$

c_s^k = cost coefficient associatedwith sub-basic variables;

c_i^θ = decision-evaluation coefficient = cost coefficient associated with θ_i;

c_n^ϕ = state-evaluation coefficient = cost coefficient associated with ϕ_n;

$[D] = [d_{ni}]$ = decision matrix, $(N\times I)$ matrix;

$[D'] = [\delta_{ni}]$ = transformed decision matrix, $(N\times I)$ matrix;

$[G_\theta]$ = decision-constraint matrix, $(L_T\times I)$ matrix;

$[G_\theta'] = \begin{bmatrix} W \\ g^\theta \end{bmatrix} = [\lambda_{si}^k]$ = transformed decision-constraint matrix, $(L_T\times I)$ matrix;

$[G_\phi]$ = state-constraint matrix, $(L_T\times N)$ matrix;

$[g^\theta]$ = sub-decision-constraint matrix, $(I\times I)$ unit matrix;

$[g^\phi]$ = sub-state-constraint matrix, $(L\times N)$ matrix;

I = total number of decision variables (total number of controllable loads);

$i = 1 \sim I$ = decision variable number (controllable load number);

i^* = decision variable number entering basis in next cycle;

j = nodal (mesh) point number associated with ith decision variable;

k = cycle number of simplex tableau;

L = total number of regulated nodal (mesh) points in water quality requirements;

L_T = total number of constraints;

$l = 1 \sim L$ = water quality requirement number;

m = nodal (mesh) point number associated with lth water quality requirement;

N = total number of state variables (total number of nodal (mesh) points);

$n, p = 1 \sim N$ = state variable number (nodal (mesh) point number);

${}_jQ_i^c = {}_j\theta_i = \theta_i$ = ith controllable load issued from nodal (mesh) point j (ith decision variable);

${}_j\overline{Q}_i^c$ = upper limit of ith controllable load;

Q_n^u = uncontrollable load at nodal (mesh) point n;

s^* = constraint number whose basic variable is replaced in next cycle;

$s = 1 \sim L_T$ = constraint number;

$[W] = - [g^\phi] \times [D']$, $(L \times I)$ matrix;

Z = objective function;

$z_i = \sum_{s=1}^{L_T} c_s^k \lambda_{si}^k$;

$\{\beta_s^k\}$ = constant vector of constraints in k cycle simplex tableau;

$\lambda_{s^* i^*}^k$ = pivot element in k cycle;

ζ_i = optimality criteria (simplex criteria);

μ_l, ν_i = slack variables;

$\xi_{s\ell} = \underset{s}{Max.}\ [\beta_s^k / \lambda_{s\vartheta*}^k],\ \lambda_{s\vartheta*}^k > 0$;

$\{\theta_i\}$ = vector of decision variables;

${}_m\overline{\Phi}_l$ = lth water quality requirement;

$\{\phi_n\}$ = vector of state variables;

$\chi_n, \chi_l^\phi, \chi_i^\theta$ = artificial variables;

$R = (L_T \times (I+L_T))$

$R' = (L \times (I+L_T)$

$R'' = (I \times (I+L_T)$.

$i = 1 \sim I + L_T$

Notes. Transformation of Equilibrium Equations by Gaussian Elimination

(1). Gaussian Elimination - Forward Reduction

The equilibrium equations (Eq. 7) are rewritten in the following form:

$$a_{11}\phi_1 + a_{12}\phi_2 + a_{13}\phi_3 \dots + a_{1N}\phi_N + d_{11}\theta_1 + d_{12}\theta_2 \dots + d_{1I}\theta_I = b_1 \quad \text{(A1-1)}$$

$$a_{21}\phi_1 + a_{22}\phi_2 + a_{23}\phi_3 \dots + a_{2N}\phi_N + d_{21}\theta_1 + d_{22}\theta_2 \dots + d_{2I}\theta_I = b_2 \quad \text{(A1-2)}$$

$$a_{31}\phi_1 + a_{32}\phi_2 + a_{33}\phi_3 \dots + a_{3N}\phi_N + d_{31}\theta_1 + d_{32}\theta_2 \dots + d_{3I}\theta_I = b_3 \quad \text{(A1-3)}$$

....

$$a_{N1}\phi_1 + a_{N2}\phi_2 + a_{N3}\phi_3 \dots + a_{NN}\phi_N + d_{N1}\theta_1 + d_{N2}\theta_2 \dots + d_{NI}\theta_I = b_N \quad \text{(A1-N)}$$

The first step in the solution of the above set of equations is to solve Eq. A1-1 for ϕ_1:

$$\phi_1 = b_1/a_{11} - (a_{12}/a_{11})\phi_2 - (a_{13}/a_{11})\phi_3 \dots - (a_{1N}/a_{11})\phi_N$$

$$- (d_{11}/a_{11})\theta_1 - (d_{12}/a_{11})\theta_2 \dots - (d_{1I}/a_{11})\theta_I \quad \text{(A2)}$$

If Eq. A-2 is substituted into Eqs. (A1-2, 3, ..., N), a modified set of $(N-1)$ equations is obtained.

$$a^1_{22}\phi_2 + a^1_{23}\phi_3 \dots + a^1_{2N}\phi_N + d^1_{21}\theta_1 + d^1_{22}\theta_2 \dots + d^1_{2I}\theta_I = b^1_2 \quad \text{(A3-2)}$$

$$a^1_{32}\phi_2 + a^1_{33}\phi_3 \dots + a^1_{3N}\phi_N + d^1_{31}\theta_1 + d^1_{32}\theta_2 \dots + d^1_{3I}\theta_I = b^1_3 \quad \text{(A3-3)}$$

....

$$a^1_{N2}\phi_2 + a^1_{N3}\phi_3 \dots + a^1_{NN}\phi_N + d^1_{N1}\theta_1 + d^1_{N2}\theta_2 \dots + d^1_{NI}\theta_I = b^1_N \quad \text{(A3-N)}$$

where

$$a^1_{np} = a_{np} - a_{n1}\, a_{1p}/a_{11} \qquad n,\ p = 2, \dots, N \quad \text{(A4)}$$

$$d_{ni}^{1} = d_{ni} - a_{n1}\, d_{1i}/a_{11} \qquad n = 2, \ldots, N;\ i = 1, 2, \ldots I \qquad \text{(A5)}$$

$$b_{n}^{1} = b_{n} - a_{n1}\, b_{1}/a_{11} \qquad n = 2, \ldots N \qquad \text{(A6)}$$

A similar procedure is used to eliminate ϕ_2 from Eq. (A3-3), etc.

A general algorithm for the elimination of ϕ_k may be written as

$$\phi_k = E_k - \sum_{p=k+1}^{N} H_{kp}\, \phi_p - \sum_{i=1}^{I} D_{ki}\, \theta_i \qquad \text{(A7)}$$

$$a_{np}^{k} = a_{np}^{k-1} - a_{nk}^{k-1}\, H_{kp} \qquad n, p = k+1, \ldots, N \qquad \text{(A8)}$$

$$d_{ni}^{k} = d_{ni}^{k-1} - a_{nk}^{k-1}\, D_{ki} \qquad n = k+1, \ldots, N;\ i = 1, 2, \ldots, I \qquad \text{(A9)}$$

$$b_{n}^{k} = b_{n}^{k-1} - a_{nk}^{k-1}\, E_k \qquad n = k+1, \ldots N \qquad \text{(A10)}$$

where

$$H_{kp} = a_{kp}^{k-1}/a_{kk}^{k-1}, \quad D_{ki} = d_{ki}^{k-1}/a_{kk}^{k-1}, \quad E_k = b_{k}^{k-1}/a_{kk}^{k-1} \qquad \text{(A11)}$$

After the above procedure is applied $(N-1)$ times the original set of equations is reduced to the following single equation.

$$a_{NN}^{N-1}\, \phi_N + \sum_{i=1}^{I} d_{Ni}^{N-1}\, \theta_i = b_{N}^{N-1} \qquad \text{(A12)}$$

which is solved directly for ϕ_N.

$$\phi_N = b_{N}^{N-1}/a_{NN}^{N-1} - \sum_{i=1}^{I} d_{Ni}^{N-1}/a_{NN}^{N-1}\, \theta_i = E_N - \sum_{i=1}^{I} D_{Ni}\, \theta_i \qquad \text{(A13)}$$

(2). Gaussian Elimination - Backward Substitution

The remaining unknowns are determined in reverse order by the repeated substitution of Eq. A 7. Therefore, a general algorithm for the solution of ϕ_k may be written as follows;

$$\phi_k = E_k - \sum_{p=k+1}^{N} H_{kp}\,\phi_p - \sum_{i=1}^{I} D_{ki}\,\theta_i$$

$$= B_k - \sum_{i=1}^{I} \delta_{ki}\,\theta_i \qquad k = N-1,\ N-2,\ \ldots,\ 1 \tag{A14}$$

where

$$B_k = E_k - \sum_{p=k+1}^{N} H_{kp}\,B_p \tag{A15}$$

$$\delta_{ki} = D_{ki} - \sum_{p=k+1}^{N} H_{kp}\,\delta_{pi} \tag{A16}$$

Thus, we obtain the transformed equilibrium equations (Eq. 17) as

$$[\,\mathrm{AI}\,]\{\phi_n\} = \{B_n\} - [\,\mathrm{D'}\,]\{\theta_i\} \tag{A17}$$

where

$$\underset{(N\times N)}{[\,\mathrm{AI}\,]} = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 1 \end{bmatrix}, \qquad \underset{(N\times I)}{[\,\mathrm{D'}\,]} = \begin{bmatrix} \delta_{11} & \delta_{12} & \cdots & \delta_{1I} \\ \delta_{21} & \delta_{22} & \cdots & \delta_{2I} \\ \delta_{31} & \delta_{32} & \cdots & \delta_{3I} \\ \cdots & \cdots & \cdots & \cdots \\ \delta_{N1} & \delta_{N2} & \cdots & \delta_{NI} \end{bmatrix} \tag{A18}$$

(3). Numerical Example

The application of the Gaussian Elimination to Eq. 28 in the numerical example yields the following Tables. (See Tables 4-A(a)-4-A(l)).

Table 4-A(a). Gaussian Elimination (Forward Reduction – k = 0)

n	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	θ_1	θ_2	b_n
1	1.21	-0.40	0.00	-0.40	0.00	0.00	0.00	0.00	2.00
2	-0.40	2.83	-0.40	0.21	-0.80	0.21	-1.00	0.00	0.00
3	0.00	-0.40	1.21	0.00	0.00	-0.40	0.00	0.00	2.00
4	-0.40	0.21	0.00	1.42	-0.40	0.00	0.00	0.00	0.00
5	0.00	-0.80	0.00	-0.40	2.42	-0.40	0.00	-1.00	0.00
6	0.00	0.21	-0.40	0.00	-0.40	1.42	0.00	0.00	0.00

Table 4-A(b). Gaussian Elimination (Forward Reduction – k = 1)

n	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	θ_1	θ_2	b_n
1	1.00	-0.33	0.00	-0.33	0.00	0.00	0.00	0.00	1.66
2	0.00	2.70	-0.40	0.08	-0.80	0.21	-1.00	0.00	0.66
3	0.00	-0.40	1.21	0.00	0.00	-0.40	0.00	0.00	2.00
4	0.00	0.08	0.00	1.30	-0.40	0.00	0.00	0.00	0.66
5	0.00	-0.80	0.00	-0.40	2.42	-0.40	0.00	-1.00	0.00
6	0.00	0.21	-0.40	0.00	-0.40	1.42	0.00	0.00	0.00

Table 4-A(c). Gaussian Elimination (Forward Reduction – k = 2)

n	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	θ_1	θ_2	b_n
1	1.00	-0.33	0.00	-0.33	0.00	0.00	0.00	0.00	1.66
2	0.00	1.00	-0.15	0.03	-0.29	0.08	-0.37	0.00	0.24
3	0.00	0.00	1.15	0.01	-0.12	-0.37	-0.15	0.00	2.10
4	0.00	0.00	0.01	1.29	-0.37	-0.01	0.03	0.00	0.64
5	0.00	0.00	-0.12	-0.37	2.19	-0.34	-0.29	-1.00	0.19
6	0.00	0.00	-0.37	-0.01	-0.34	1.40	0.08	0.00	-0.05

Table 4-A(d). Gaussian Elimination (Forward Reduction – k = 3)

n	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	θ_1	θ_2	b_n
1	1.00	-0.33	0.00	-0.33	0.00	0.00	0.00	0.00	1.66
2	0.00	1.00	-0.15	0.03	-0.29	0.08	-0.37	0.00	0.24
3	0.00	0.00	1.00	0.01	-0.10	-0.32	-0.13	0.00	1.82
4	0.00	0.00	0.00	1.29	-0.37	-0.00	0.03	0.00	0.62
5	0.00	0.00	0.00	-0.37	2.17	-0.37	-0.31	-1.00	0.40
6	0.00	0.00	0.00	-0.00	-0.37	1.29	0.03	0.00	0.62

Table 4-A(e). Gaussian Elimination (Forward Reduction – k = 4)

n	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	θ_1	θ_2	b_n
1	1.00	-0.33	0.00	-0.33	0.00	0.00	0.00	0.00	1.66
2	0.00	1.00	-0.15	0.03	-0.29	0.08	-0.37	0.00	0.24
3	0.00	0.00	1.00	0.01	-0.10	-0.32	-0.13	0.00	1.82
4	0.00	0.00	0.00	1.00	-0.29	-0.00	0.02	0.00	0.48
5	0.00	0.00	0.00	0.00	2.07	-0.37	-0.30	-1.00	0.58
6	0.00	0.00	0.00	0.00	-0.37	1.29	0.03	0.00	0.62

Table 4-A(f). Gaussian Elimination (Forward Reduction – k = 5)

n	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	θ_1	θ_2	b_n
1	1.00	-0.33	0.00	-0.33	0.00	0.00	0.00	0.00	1.66
2	0.00	1.00	-0.15	0.03	-0.29	0.08	-0.37	0.00	0.24
3	0.00	0.00	1.00	0.01	-0.10	-0.32	-0.13	0.00	1.82
4	0.00	0.00	0.00	1.00	-0.29	-0.00	0.02	0.00	0.48
5	0.00	0.00	0.00	0.00	1.00	-0.18	-0.14	-0.48	0.28
6	0.00	0.00	0.00	0.00	0.00	1.22	-0.02	-0.18	0.72

Table 4-A(g). Gaussian Elimination (Forward Reduction – k = 6) (Backward Substitution – k = 6)

n	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	θ_1	θ_2	B_n
1	1.00	-0.33	0.00	-0.33	0.00	0.00	0.00	0.00	1.66
2	0.00	1.00	-0.15	0.03	-0.29	0.08	-0.37	0.00	0.24
3	0.00	0.00	1.00	0.01	-0.10	-0.32	-0.13	0.00	1.82
4	0.00	0.00	0.00	1.00	-0.29	-0.00	0.02	0.00	0.48
5	0.00	0.00	0.00	0.00	1.00	-0.18	-0.14	-0.48	0.28
6	0.00	0.00	0.00	0.00	0.00	1.00	-0.02	-0.15	0.59

Table 4-A(h). Gaussian Elimination (Backward Substitution – k = 5)

n	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	θ_1	θ_2	B_n
1	1.00	-0.33	0.00	-0.33	0.00	0.00	0.00	0.00	1.66
2	0.00	1.00	-0.15	0.03	-0.29	0.08	-0.37	0.00	0.24
3	0.00	0.00	1.00	0.01	-0.10	-0.32	-0.13	0.00	1.82
4	0.00	0.00	0.00	1.00	-0.29	-0.00	0.02	0.00	0.48
5	0.00	0.00	0.00	0.00	1.00	0.00	-0.15	-0.51	0.39
6	0.00	0.00	0.00	0.00	0.00	1.00	-0.02	-0.15	0.59

Table 4-A(i). Gaussian Elimination (Backward Substitution – k = 4)

n	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	θ_1	θ_2	B_n
1	1.00	-0.33	0.00	-0.33	0.00	0.00	0.00	0.00	1.66
2	0.00	1.00	-0.15	0.03	-0.29	0.08	-0.37	0.00	0.24
3	0.00	0.00	1.00	0.01	-0.10	-0.32	-0.13	0.00	1.82
4	0.00	0.00	0.00	1.00	0.00	0.00	-0.02	-0.15	0.59
5	0.00	0.00	0.00	0.00	1.00	0.00	-0.15	-0.51	0.39
6	0.00	0.00	0.00	0.00	0.00	1.00	-0.02	-0.15	0.59

Table 4-A(j). Gaussian Elimination (Backward Substitution – k = 3)

n	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	θ_1	θ_2	B_n
1	1.00	-0.33	0.00	-0.33	0.00	0.00	0.00	0.00	1.66
2	0.00	1.00	-0.15	0.03	-0.29	0.08	-0.37	0.00	0.24
3	0.00	0.00	1.00	0.00	0.00	0.00	-0.15	-0.10	2.04
4	0.00	0.00	0.00	1.00	0.00	0.00	-0.02	-0.15	0.59
5	0.00	0.00	0.00	0.00	1.00	0.00	-0.15	-0.51	0.39
6	0.00	0.00	0.00	0.00	0.00	1.00	-0.02	-0.15	0.59

Table 4-A(k). Gaussian Elimination (Backward Substitution – k = 2)

n	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	θ_1	θ_2	B_n
1	1.00	-0.33	0.00	-0.33	0.00	0.00	0.00	0.00	1.66
2	0.00	1.00	0.00	0.00	0.00	0.00	-0.43	-0.15	0.59
3	0.00	0.00	1.00	0.00	0.00	0.00	-0.15	-0.10	2.04
4	0.00	0.00	0.00	1.00	0.00	0.00	-0.02	-0.15	0.59
5	0.00	0.00	0.00	0.00	1.00	0.00	-0.15	-051	0.39
6	0.00	0.00	0.00	0.00	0.00	1.00	-0.02	-0.15	0.59

Table 4-A(l). Gaussian Elimination (Backward Substitution – k = 1)

n	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	θ_1	θ_2	B_n
1	1.00	0.00	0.00	0.00	0.00	0.00	-0.15	-0.10	2.04
2	0.00	1.00	0.00	0.00	0.00	0.00	-0.43	-0.15	0.59
3	0.00	0.00	1.00	0.00	0.00	0.00	-0.15	-0.10	2.04
4	0.00	0.00	0.00	1.00	0.00	0.00	-0.02	-0.15	0.59
5	0.00	0.00	0.00	0.00	1.00	0.00	-0.15	-0.51	0.39
6	0.00	0.00	0.00	0.00	0.00	1.00	-0.02	-0.15	0.59

Chapter 5

TRANSIENT FINITE ELEMENT & LINEAR PROGRAMMING METHOD IN WATER POLLUTION CONTROL

Summary

A transient finite element & linear programming method (T. FELP Method) is developed in order to control systems of unsteady state differntial equations with both equality or inequality constraints and an objective function. Such systems are frequently encountered in various engineering and scientific problems of control and optimal design. The applicability of T. FELP Method is shown through the control of diffusion-convection phenomena in water pollution problems. The tractability in the initial and boundary conditions and in the equality or inequality constraints makes sure that T. FELP Method becomes one of the useful techniques for several new types of boundary value problems. T. FELP Method may become an useful technique for analysis, planning and assessment in environmental and water resources problems.

5-1. General Concepts

By combining finite element method with linear programming, a transient finite element & linear programming method (T. FELP Method) is developed in order to solve systems of unsteady state differential equations with both equality or inequality constraints and an objective function. Such systems are frequently encountered in various engineering and scientific problems of control and optimal design and, especially, are of interest in environmental and water resources problems. Research on finite element & linear programming method (FELP Method) in control problems of steady state field phenomena has been presented (7). (See Chapter 3).

The finite element method, originated in structural mechanics, is powerful numerical method for the solution of differential equations because of its generality with respect to geometry and material properties (3, 4, 14, 18, 19). Moreover, linear programming is one of the most frequently used mathematical methods of operations research (5, 9).

In the development of T. FELP Method the concepts of the decision variable and the state variable are adopted as in Bellman's dynamic programming (2, 8) and Pontryagin's maximum principle (6, 13). T. FELP Method utilizes the advantages of the established numerical techniques of both finite element method and linear programming. The various problems formulated by T. FELP Method could be solved by the combined use of existing computer codes for finite element method and linear programming.

The applicability of T. FELP Method is shown through the control of diffusion-convection phenomena in water pollution problems. In the application of T. FELP Method, researches on water pollution problems associated with diffusion phenomena have provided the stimulas and many useful ideas (10, 11, 12, 15, 16, 17).

5-2. Transient Finite Element & Linear Programming Method

5-2-1. Systems of Basic Differential Equations

Transient finite element & linear programming method (T. FELP Method) is developed in order to control the following systems of differential equations (See Fig. 5-1).

Objective Function (throughout the whole domain ($\Omega = \Omega_s \times \Omega_t$))

$$Z = \underset{\{\{\theta\}\}}{Opt.}\ f\ (\{\{\phi\}\},\ \{\{\theta\}\}) = \begin{cases} \underset{\{\{\theta\}\}}{Max.}\ f\ (\{\{\phi\}\},\ \{\{\theta\}\}) \\ \underset{\{\{\theta\}\}}{Min.}\ f\ (\{\{\phi\}\},\ \{\{\theta\}\}) \end{cases} \tag{1}$$

subject to:

Equilibrium Equations

Governing Equation (in the whole domain ($\Omega = \Omega_s \times \Omega_t$))

$$D.E.\ (x_k,\ t,\ \phi,\ \theta) = 0 \tag{2}$$

Boundary Conditions (on the boundaries S, in the whole time domain Ω_t)

$$h\ (X_k,\ t,\ \phi) = 0 \tag{3}$$

Initial Condition (in the whole space domain Ω_s, at the time $t = 0$)

$$e\ (x_k,\ t=0,\ \phi) = 0 \tag{4}$$

Constraints (in the subdomains ($\Omega^g = \Omega^g_s \times \Omega^g_t$))

$$g\ (x_k,\ t,\ \phi,\ \theta) \lesseqgtr 0 \tag{5}$$

in which $\phi = \phi(x_k,\ t)$ = the state variable; $\theta = \theta(x_k,\ t)$ = the decision variable; $\{\{\phi\}\}$ = vector of the state variables in the whole domain Ω; $\{\{\theta\}\}$ = vector of the decision variables in the whole domain Ω; x_k =

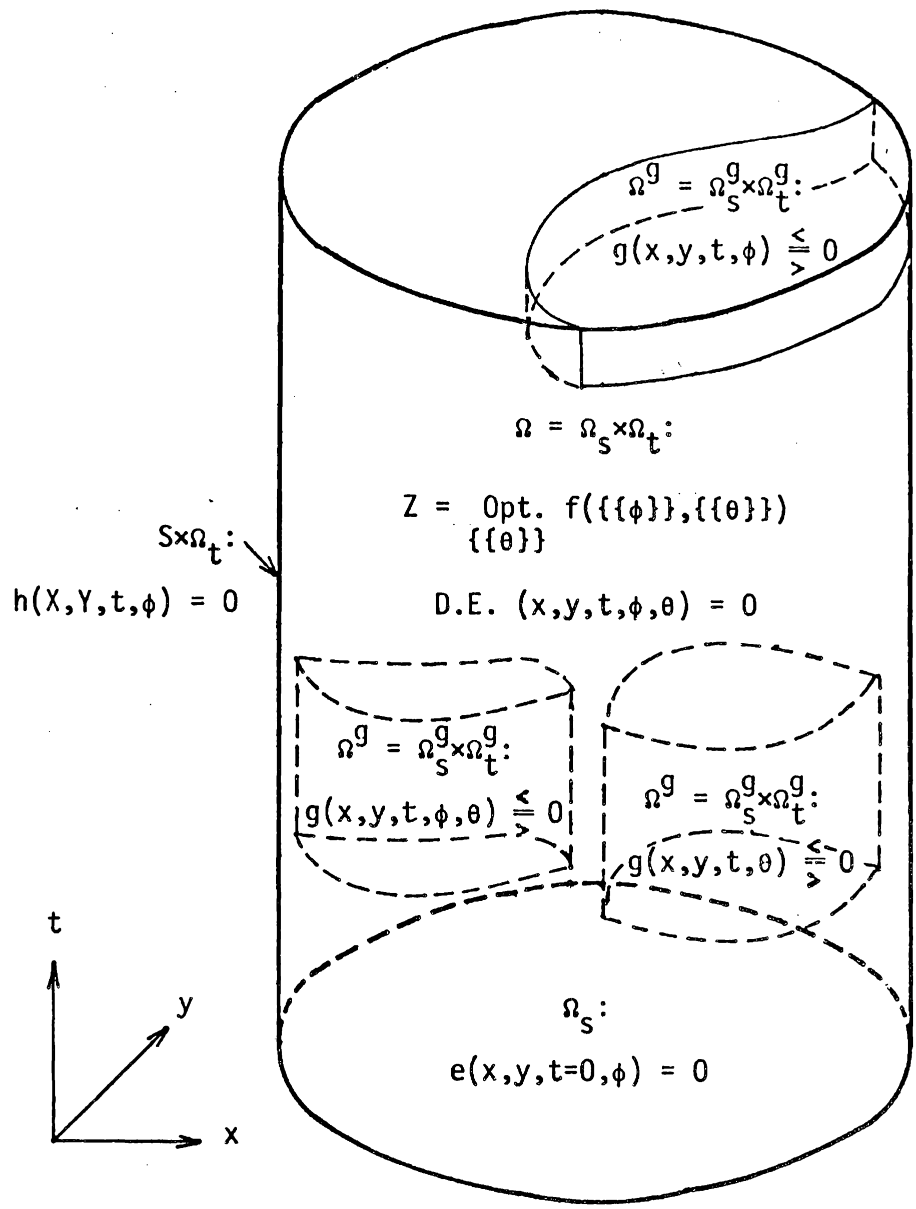


Fig. 5-1. General Concepts of Transient Finite Element & Programming Method

Cartesian coordinate (x, y or z); t = time; and X_k = Cartesian coordinate of the boundary (X, Y or Z).

In systems governed by three dimensional quasi-linear parabolic equation, for example, Eq. 2 is expressed as follows:

$$\underbrace{c_1 \frac{\partial \phi}{\partial t} = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2}}_{\phi\text{-}terms} + \underbrace{c_2\, \theta}_{\theta\text{-}term} + \underbrace{b}_{const} \tag{2}'$$

The examples of the boundary conditions are as follows:

$$\phi(X,Y,Z,t) = \Phi_b(X,Y,Z,t), \quad \frac{\partial \phi}{\partial n}\Big|(X,Y,Z,t) = 0 \tag{3}'$$

The example of the initial condition is as follows:

$$\phi(x,y,z,t=0) = \Phi^0(x,y,z) \tag{4}'$$

As for the examples of the constraints, the following simple inequality are frequently encountered.

$$\begin{aligned} &\underline{\Phi} \leqq \phi \leqq \overline{\Phi}, \ or \ \begin{cases} \phi \geqq \underline{\Phi} \\ \phi \leqq \overline{\Phi} \end{cases} \\ &\underline{\Theta} \leqq \theta \leqq \overline{\Theta}, \ or \ \begin{cases} \theta \geqq \underline{\Theta} \\ \theta \leqq \overline{\Theta} \end{cases} \end{aligned} \tag{5}'$$

in which $\underline{\Phi}$ = the lower limit of the state variable; $\overline{\Phi}$ = the upper limit of the state variable; $\underline{\Theta}$ = the lower limit of the decision variable; and $\overline{\Theta}$ = the upper limit of the decision variable.

5-2-2. Formulation of T. FELP Method

Finite element method is used in order to discretize the above-mentioned systems as systems of linear algebraic equations. (As for the details, see the next section). Then, the following matrix-vector forms of T. FELP Method are obtained and the application of linear programming is possible.

Objective Function

$$Z = \underset{\{\{{}_j\theta_i^\tau\}\}}{Opt.}\ f\,(\{\{\phi_n^\tau\}\},\ \{\{{}_j\theta_i^\tau\}\}) = \underset{\{\{{}_j\theta_i^\tau\}\}}{Opt.}\ \sum_{\tau=1}^{T} \left(\sum_{n=1}^{N} {}^{\phi}c_n^\tau\, \phi_n + \sum_{i=1}^{I} {}^{\theta}c_i^\tau\, {}_j\theta_i \right) \quad (6)$$

subject to:

Equilibrium Equations ((T×N)-Eqs.)

$$[\mathrm{A} + \frac{1}{\Delta t}\,\mathrm{C}]\{\phi_n^1\} + [\mathrm{D}]\{{}_j\theta_i^1\} = \{b_n^1\} + [\frac{1}{\Delta t}\,\mathrm{C}]\{\phi_n^0\} \qquad (\tau = 1) \qquad (7)$$

$$-[\frac{1}{\Delta t}\,\mathrm{C}]\{\phi_n^{\tau-1}\} + [\mathrm{A} + \frac{1}{\Delta t}\,\mathrm{C}]\{\phi_n^\tau\} + [\mathrm{D}]\{{}_j\theta_i^\tau\} = \{b_n^\tau\} \quad (\tau = 2 \sim T) \quad (8)$$

Constraints (L_T-Eqs.)

$$[\mathrm{G}_\phi]\{\{\phi_n^\tau\}\} + [\mathrm{G}_\theta]\{\{{}_j\theta_i^\tau\}\} \ \substack{\leq \\ = \\ >}\ \{\{b_l^g\}\} \qquad (9)$$

Nonnegative Conditions

$$\phi_n^\tau \geqq 0 \ (\tau = 1 \sim T,\ n = 1 \sim N),\quad {}_j\theta_i^\tau \geqq 0 \ (\tau = 1 \sim T,\ i = 1 \sim I) \quad (10)$$

in which [A] = the state matrix (the global stiffness matrix), $(N\times N)$ matrix; [C] = the capacity matrix, $(N\times N)$ matrix; $[\mathrm{G}_\phi]$ = the state-constraint matrix, $(L_T\times(T\times N))$ matrix; $[\mathrm{G}_\theta]$ = the decision-constraint matrix, $(L_T\times(T\times I))$ matrix; $\{\{\phi_n^\tau\}\}$ = vector of the state variables in the whole domain Ω; $\{\{{}_j\theta_i^\tau\}\}$ = vector of the decision variables in the whole domain Ω; $\{\phi_n^\tau\}$ = vector of the state variables at the time step τ $\{{}_j\theta_i^\tau\}$ = vector of the decision variables at the time step τ; $\{\Phi_n^0\}$ = vector of the initial states; $\{b_n^\tau\}$ = vector of the constants at the time step τ; $\{\{b_l^g\}\}$ = vector of the constants in the constraints; ${}^\phi c_n^\tau$ = state-evaluation coefficient = cost coefficient associated with ϕ_n^τ; ${}^\theta c_i^\tau$ = decision-evaluation coefficient = cost coefficient associated with ${}_j\theta_i^\tau$; $\tau = 1 \sim T$ = time step number; T = total number of the time steps; Δt = increment in time; $n = 1 \sim N$ = state variable number at each time step (nodal point number in finite elements); N = number of the state variables at each time step (total number of nodal points in finite elements); i = $1 \sim I$ = decision variable number at each time step; I = number of the decision variables at each time step; j = nodal point number associated with ith decision variable; $l = 1 \sim L_T$ = constraint number; and L_T = total number of the constraints.

T. FELP Method is one that optimizes the objective function under the conditions of the equilibrium equations and the constraints. Since all of the variables in linear programming have to be nonnegative because of the limitation in the computational algorithm based on the simplex method, the nonnegative conditions (Eq. 10) are required. In T. FELP Method the number of the variables is $(T\times(N+I))$, the number of the equilibrium equations is $(T\times N)$, and the number of the constraints is L_T, respectively. In the sense of general linear programming, the equilibrium equations of T. FELP Method are also the constraints. Thus, T. FELP Method is a kind of linear programming in which the number of the variables is $(T\times(N+I))$ and the number of the constraints is $((T\times N)+L_T))$, respectively. In T. FELP Method the solution for the state variables and the solution for the decision variables are obtained simultaneously by the simplex method.

5-3. Water Pollution Control by T. FELP Method

5-3-1. Systems of Basic Equations in Diffusion-Convection Phenomena

The basic equation systems of the unsteady state two-dimensional diffusion-convection phenomena in water pollution problems are as follows:

Objective Function (throughout the whole domain ($\Omega = \Omega_s \times \Omega_t$))

$$Z = \underset{\{\{\theta\}\}}{Opt.}\ f\ (\{\{\phi\}\},\ \{\{\theta\}\}) \simeq \underset{\{\{\theta\}\}}{Max.}\ \Sigma\ \theta \tag{11}$$

subject to:

Equilibrium Equations

Governing Equation (in the whole domain ($\Omega = \Omega_s \times \Omega_t$))

$$\underbrace{\frac{\partial \phi}{\partial t} = \sum_{k=1}^{2} \left(\frac{\partial}{\partial x_k} D_{xk} \frac{\partial \phi}{\partial x_k} - v_k \frac{\partial \phi}{\partial x_k}\right) - K\ \phi}_{\phi\text{-terms}} + \underbrace{\theta}_{\theta\text{-term}} + \underbrace{Q}_{const} \tag{12}$$

Boundary Conditions (on the boundaries S, in the whole time domain Ω_t)

$$\phi(X,Y,t) = \Phi_b(X,Y,t) \qquad \text{on } S^1 \tag{13}$$

$$\sum_{k=1}^{2} D_{xk} \frac{\partial \phi}{\partial x_k} l_{xk} = -\ q - \kappa\ (\phi - \Phi_a) \qquad \text{on } S^2 \tag{14}$$

Initial Condition (in the whole water basin Ω_s, at the time $t = 0$)

$$\phi(x,y,t=0) = \Phi^0(x,y) \tag{15}$$

Constraints (in the subdomains ($\Omega^g = \Omega_s^g \times \Omega_t^g$))

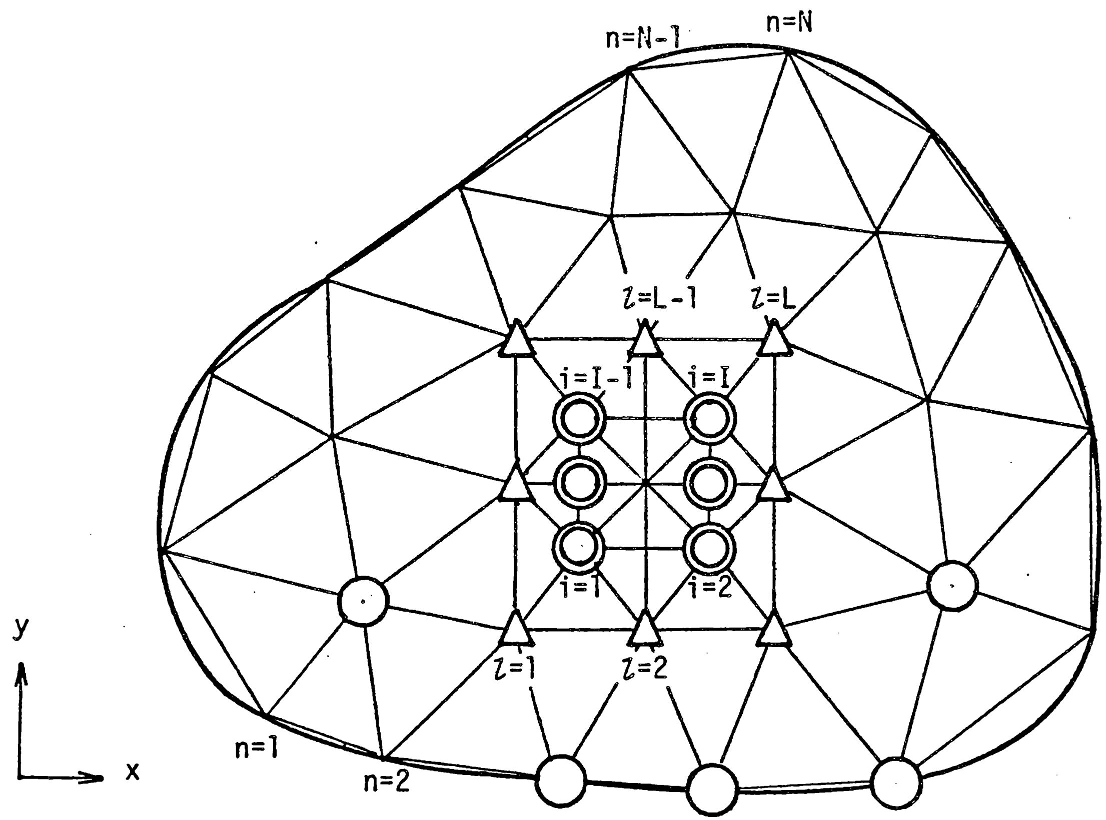
$$\phi \leqq \overline{\Phi} \tag{16}$$

$$\theta \leqq \overline{\Theta} \tag{17}$$

in which ϕ = the state variable = water quality in the water basin (e.g., temperature in thermal pollution problem or concentration of pollutant); θ = the decision variable = controllable load (e.g., heated discharge from multi-port diffuser or pollutant issued from waste outfalls); Q = constant = uncontrollable load (inevitably or naturally generated source or sink, existing unexcludable discharge); D_{xk} = diffusion coefficient (D_x or D_y); v_k = convective velocity (v_x or v_y); K = decay factor (e.g., heat transfer coefficient at the water surface or decay rate of pollutant); S^1 and S^2 = the parts of the boundary S; Φ_b = prescribed boundary value; l_{xk} = the direction cosine of the outward normal to the boundary (l_x or l_y); q = intensity of flux per unit length of the surrounding boundary; κ = decay factor of the surrounding boundary; Φ_a = temperature or concentration of the surrounding boundary; $\overline{\Phi}$ = water quality requirement; and $\overline{\Theta}$ = the upper limit of the controllable load.

If D_x and D_y are equal and both q and κ are equal to zero, a well-known condition applicable to non-conductive boundaries is obtained, that is, $\partial\phi/\partial n = 0$.

Although, the objective function may be composed of the water quality distribution $\{\{\phi\}\}$ and the controllable loads $\{\{\theta\}\}$ in general, the maximization of the total of the controllable loads in the whole domain Ω is sought in the numerical example for simplicity. From the view point of the assimilation capacity of the water basin, such an objective function gives us the upper limit of the total acceptable load in the water basin. As for the constraints, although only the upper limits of the state variable and the decision variables are imposed, we can impose other constraints, if necessary. Therefore, the problem to be solved is one that seeks not only the optimal controllable loads to meet water quality requirements but also the distribution patterns of the water quality.



◎ : Nodal point fitted for outfall of controllable load

○ : Nodal point fitted for outfall of uncontrollable load

△ : Regulated nodal point in water quality requirement

Fig. 5-2. Water Basin Divided into Triangular Elements

5-3-2. Formulation of T. FELP Method in Diffusion-Convection Phenomena

In order to discretize the systems of differential equations (Eqs. 11-17), Galerkin finite element method is used because of its independency of variational principle. As for the details of Galerkin finite element method, one may follow Zienkiewicz and use the shape functions as the weighting functions (19). The water basin to be analyzed is divided into small regions called finite elements. (See Figs. 5-2 and 5-3). Let the state variable be approximated, throughout the whole space domain Ω_s (throughout the whole water basin) at the time step τ, by the following relationship:

$$\phi = [\mathrm{N}]\{\phi_n^\tau\} = \sum_{n=1}^{N} N_n(x,y)\ \phi_n^\tau \tag{18}$$

in which $[\mathrm{N}] = [N_1 \ .. \ N_n \ .. \ N_n]$ = the usual shape functions defined piecewise, element by element; and $\{\phi_n^\tau\}$ = the nodal parameters at the time step τ.

Using the weighted residual process, the Galerkin representation for the diffusion-convection equation (Eq. 12) is as follows:

$$\int_\Delta N_n \left[\sum_{k=1}^{2} \left(\frac{\partial}{\partial x_k} D_{xk} \frac{\partial \phi}{\partial x_k} - v_k \frac{\partial \phi}{\partial x_k} \right) - K\ \phi + \theta + Q - \frac{\partial \phi}{\partial t} \right] d\Delta \ = \ 0 \tag{19}$$

in which Δ = area of each element.

Using integration by parts on the first and second terms in the above equation and inserting the boundary condition (Eq. 14), we obtain the following equation.

$$- \int_\Delta \left[\sum_{k=1}^{2} D_{xk} \frac{\partial N_n}{\partial x_k} \frac{\partial \phi}{\partial x_k} + \sum_{k=1}^{2} v_k\ N_n \frac{\partial \phi}{\partial x_k} + K\ N_n\ \phi - \theta\ N_n - Q\ N_n + N_n \frac{\partial \phi}{\partial t} \right] d\Delta$$

$$- \int_{s^2} N_n \left[q + \kappa\ (\phi - \Phi_a) \right] ds^2 \ = 0 \tag{20}$$

in which s^2 = segment of the part of the boundary S^2 refering only to elements with external boundaries on which the boundary condition (Eq. 14) is specified.

Substituting Eq. 18 into the above equation, the summation being taken all over the elements yields the N-linear algebraic equations at each time step

$$[A]\{\phi_n^\tau\} + [C]\{\frac{\partial\phi}{\partial t}\Big|_n^\tau\} - \{\theta_n^\tau\} = \{Q_n^\tau\} \quad (\tau = 1 \sim T) \tag{21}$$

with the following matrix and vector elements

$$a_{np} = \Sigma \int_\Delta \Big[\sum_{k=1}^{2} D_{xk} \frac{\partial N_n}{\partial x_k}\frac{\partial N_p}{\partial x_k} + \sum_{k=1}^{2} v_k N_n \frac{\partial N_p}{\partial x_k} + K N_n N_p \Big] d\Omega_s$$

$$+ \Sigma \int_{s^2} \kappa N_n N_p \, dS^2 \tag{22}$$

$$c_{np} = \Sigma \int_\Delta N_n N_p \, d\Omega_s \tag{23}$$

$$\theta_n^\tau = \Sigma \int_\Delta \theta N_n \, d\Omega_s \tag{24}$$

$$Q_n^\tau = \Sigma \int_\Delta Q N_n \, d\Omega_s - \Sigma \int_{s^2} (q - \kappa \Phi_a) N_n \, dS^2 \tag{25}$$

in which $[A] = [a_{np}]$ = the state matrix = the global stiffness matrix (the diffusiveness matrix), $(N{\times}N)$ matrix; $[C] = [c_{np}]$ = the capacity matrix, $(N{\times}N)$ matrix; ϕ_n^τ = nth state variable at the time step τ (water quality at the nodal point n and at the time step τ; $\frac{\partial\phi}{\partial t}\Big|_n^\tau$ = rate of change of nth state variable at the time step τ; θ_n^τ = nth decision variable at the time step τ (controllable load at the nodal point n and at the time step τ); Q_n^τ = constant = uncontrollable load at the nodal point n and at the time step τ; and n, p = $1 \sim N$ = subscripts.

In order to reduce the number of the decision variables, θ_n^τ should be dropped at the nodal points where the outfall for the controllable load does not exist. Therefore, the following expression for the controllable loads is used instead of $\{\theta_n^\tau\}$ and the number of the decision variables at each time step is reduced from N to I.

$$- [D]\{{}_j\theta_i^\tau\} = - [d_{ni}]\{{}_j\theta_i^\tau\} \tag{26}$$

in which $[D] = [d_{ni}]$ = the decision matrix, $(N\times I)$ matrix composed of zero elements with the exceptions of '-1' in I elements whose row number is j and whose column number is i, (see Eq. 47); ${}_j\theta_i^\tau$ = ith decision variable (ith controllable load) at the time step τ; $i = 1 \sim I$ = decision variable number at each time step (controllable load number at each time step); I = number of the decision variables at each time step (number of the controllable loads, i.e., total number of the nodal points fitted for the locations of the outfalls of the controllable loads); and j = nodal point number fitted for the location of ith controllable load.

Using the above expression for the controllable loads, Eq. 21 is rewritten as follows:

$$[A]\{\phi_n^\tau\} + [C]\{\frac{\partial\phi}{\partial t}\Big|_n^\tau\} + [D]\{{}_j\theta_i^\tau\} = \{Q_n^\tau\} \quad (\tau = 1 \sim T) \tag{27}$$

Although several time stepping schemes in finite element method have been presented, the following backward differencing (18) is used in this research.

$$\{\frac{\partial\phi}{\partial t}\Big|_n^\tau\} = \frac{1}{\Delta t}\left(\{\phi_n^\tau\} - \{\phi_n^{\tau-1}\}\right) \quad (\tau = 1 \sim T) \tag{28}$$

Substitution of the above equation into Eq. 27 yields the following N-equilibrium equations at each time step.

$$- [\frac{1}{\Delta t} C]\{\phi_n^{\tau-1}\} + [A + \frac{1}{\Delta t} C]\{\phi_n^{\tau}\} + [D]\{{}_j\theta_i^{\tau}\} = \{Q_n^{\tau}\} \quad (\tau = 1 \sim T) \quad (29)$$

At the nodal points on the boundary S^1 with the prescribed boundary value of $\phi(X,Y,t) = \Phi_b(X,Y,t)$ (Eq. 13), the equilibrium equation is corrected as follows:

$$a_{nn}\, \phi_n^{\tau} = \Phi_b^{\tau}, \ or \quad 1 \times \phi_n^{\tau} = \Phi_b^{\tau} \quad (30)$$

Substituting the initial condition (Eq. 15) into Eq. 29, we obtain the following N-equilibrium equations at the time step $\tau = 1$.

$$[A + \frac{1}{\Delta t} C]\{\phi_n^{1}\} + [D]\{{}_j\theta_i^{1}\} = \{Q_n^{1}\} + [\frac{1}{\Delta t} C]\{\Phi_n^{0}\} \quad (\tau = 1) \quad (31)$$

in which Φ_n^0 = the initial state at the nodal point n (the initial water quality at the nodal point n).

Thus, the discretized equilibrium equations are obtained.

The constraints associated with the water quality requirement (Eq. 16) are discretized at each time step as follows:

$$[g^{\phi}]\{\phi_n^{\tau}\} \leqq \{{}_m\overline{\Phi}_l^{\tau}\} \quad (\tau = 1 \sim T), \ or \quad \phi_m^{\tau} \leqq {}_m\overline{\Phi}_l^{\tau} \quad (\tau = 1 \sim T,\ l = 1 \sim L) \quad (32)$$

in which ${}_m\overline{\Phi}_l^{\tau}$ = lth water quality requirement at the time step τ, i.e., the upper limit of the mth state variable at the time step τ; $l = 1 \sim L$ = water quality requirement number at each time step; L = number of regulated nodal points in water quality requirements at each time step; m = regulated nodal point number in lth water quality requirement; and $[g^{\phi}] = [g_{ln}^{\phi}]$ = the sub-state-constraint matrix, $(L{\times}N)$ matrix composed of zero elements with the exceptions of '1' in L elements whose row number is l and whose column number is m, (see Eq. 48).

The discretized constraints associated with the upper limits of the controllable loads are as follows:

$$[g^{\theta}]\{{}_j\theta_i^{\tau}\} \leqq \{{}_j\overline{\theta}_i^{\tau}\} \quad (\tau = 1 \sim T), \text{ or } \quad {}_j\theta_i^{\tau} \leqq {}_j\overline{\theta}_i^{\tau} \quad (\tau = 1 \sim T,\ i = 1 \sim I) \tag{33}$$

in which ${}_j\overline{\theta}_i^{\tau}$ = the upper limit of ith controllable load; and $[g^{\theta}] = [g_{ii}^{\theta}]$ = the sub-decision-constraint matrix, $(I\times I)$ unit matrix, (see Eq. 49).

Then, the following formulation of T. FELP Method in control of the diffusion-convection phenomena in water pollution problems is obtained.

Objective Function

$$Z = \underset{\{\{{}_j\theta_i^{\tau}\}\}}{Opt.}\ f\,(\{\{\phi_n^{\tau}\}\},\ \{\{{}_j\theta_i^{\tau}\}\}) = \underset{\{\{{}_j\theta_i^{\tau}\}\}}{Opt.}\ \sum_{\tau=1}^{T}\ (\sum_{n=1}^{N} {}^{\phi}c_n^{\tau}\,\phi_n^{\tau} + \sum_{i=1}^{I} {}^{\theta}c_i^{\tau}\,{}_j\theta_i^{\tau})$$

$$\simeq \sum_{\tau=1}^{T}\sum_{i=1}^{I} \Delta t\ {}_j\theta_i^{\tau} \tag{34}$$

subject to:

Equilibrium Equations ((T×N)-Eqs.)

$$[A + \frac{1}{\Delta t}C]\{\phi_n^{1}\} + [D]\{{}_j\theta_i^{1}\} = \{Q_n^{1}\} + [\frac{1}{\Delta t}C]\{\phi_n^{0}\} \quad (\tau = 1) \tag{35}$$

$$-[\frac{1}{\Delta t}C]\{\phi_n^{\tau-1}\} + [A + \frac{1}{\Delta t}C]\{\phi_n^{\tau}\} + [D]\{{}_j\theta_i^{\tau}\} = \{Q_n^{\tau}\} \quad (\tau = 2 \sim T) \tag{36}$$

Constraints ($(L_T = T\times(L+I))$-Eqs.)

$$[g^{\phi}]\{\phi_n^{\tau}\} \leqq \{{}_m\overline{\Phi}_l^{\tau}\} \quad (\tau = 1 \sim T) \tag{37}$$

$$[g^{\theta}]\{{}_j\theta_i^{\tau}\} \leqq \{{}_j\overline{\theta}_i^{\tau}\} \quad (\tau = 1 \sim T) \tag{38}$$

Nonnegative Conditions

$$\phi_n^{\tau} \geqq 0\ (\tau = 1 \sim T,\ n = 1 \sim N), \quad {}_j\theta_i^{\tau} \geqq 0\ (\tau = 1 \sim T,\ i = 1 \sim I) \tag{39}$$

In the maximization problem of total of the controllable loads, all of the state-evaluation coefficients, cost coefficients associated with the state variables, $\{\{{}^{\phi}c_n^{\tau}\}\}$ are equal to zero and all of the decision-evaluation coefficients, cost coefficients associated with the decision variables, $\{\{{}^{\theta}c_i^{\tau}\}\}$ are equal to Δt.

5-4. Numerical Example in a Simple Model Basin

In order to clarify the features of T. FELP Method a numerical example of the method is conducted in a rectangular model basin with non-conductive boundaries as shown in Fig. 5-3.

The units of the water quality and the loads are not described, because the method is applicable to general physical problems irrespective of the variables considered.

The obtained matrix-vector forms of T. FELP Method are as follows:

Objective Function

$$Z = \underset{\{\{{}_j\theta_i^\tau\}\}}{Opt.} \sum_{\tau=1}^{3} \left(\sum_{n=1}^{6} {}^{\phi}c_n^\tau \phi_n^\tau + \sum_{i=1}^{2} {}^{\theta}c_i^\tau \, {}_j\theta_i^\tau \right) \simeq \underset{\{\{{}_j\theta_i^\tau\}\}}{Max.} \sum_{\tau=1}^{3} \sum_{i=1}^{2} \Delta t \, {}_j\theta_i^\tau \quad (40)$$

subject to:

Equilibrium Equations ((3×6)-Eqs.)

$$[A + \frac{1}{\Delta t} C]\{\phi_n^1\} + [D]\{{}_j\theta_i^1\} = \{Q_n^1\} + [\frac{1}{\Delta t} C]\{\phi_n^0\} \quad (\tau = 1) \qquad (41\text{-}1)$$

$$-[\frac{1}{\Delta t} C]\{\phi_n^1\} + [A + \frac{1}{\Delta t} C]\{\phi_n^2\} + [D]\{{}_j\theta_i^2\} = \{Q_n^2\} \quad (\tau = 2) \qquad (41\text{-}2)$$

$$-[\frac{1}{\Delta t} C]\{\phi_n^2\} + [A + \frac{1}{\Delta t} C]\{\phi_n^3\} + [D]\{{}_j\theta_i^3\} = \{Q_n^3\} \quad (\tau = 3) \qquad (41\text{-}3)$$

Constraints ((12 = 3×(2+2))-Eqs.)

$$[g^\phi]\{\phi_n^1\} \leqq \{{}_m\bar{\Phi}_l^1\}, \text{ or } \begin{cases} \phi_4^1 \leqq {}_4\bar{\Phi}_1^1 \ (= 3.0) \\ \phi_6^1 \leqq {}_6\bar{\Phi}_2^1 \ (= 3.0) \end{cases} \qquad (42\text{-}1)$$

$$[g^\phi]\{\phi_n^2\} \leqq \{{}_m\bar{\Phi}_l^2\}, \text{ or } \begin{cases} \phi_4^2 \leqq {}_4\bar{\Phi}_1^2 \ (= 5.0) \\ \phi_6^2 \leqq {}_6\bar{\Phi}_2^2 \ (= 5.0) \end{cases} \qquad (42\text{-}2)$$

$$[g^{\phi}]\{\phi_n^3\} \leqq \{{}_m\overline{\Phi}_l^3\}, \text{ or } \begin{cases} \phi_4^3 \leqq {}_4\overline{\Phi}_1^3 \ (= 5.0) \\ \phi_6^3 \leqq {}_6\overline{\Phi}_2^3 \ (= 5.0) \end{cases} \tag{42-3}$$

$$[g^{\theta}]\{{}_j\theta_i^1\} \leqq \{{}_j\overline{\Theta}_i^1\}, \text{ or } \begin{cases} {}_2\theta_1^1 \leqq {}_2\overline{\Theta}_1^1 \ (= 3.5) \\ {}_5\theta_2^1 \leqq {}_5\overline{\Theta}_2^1 \ (= 3.5) \end{cases} \tag{43-1}$$

$$[g^{\theta}]\{{}_j\theta_i^2\} \leqq \{{}_j\overline{\Theta}_i^2\}, \text{ or } \begin{cases} {}_2\theta_1^2 \leqq {}_2\overline{\Theta}_1^2 \ (= 3.5) \\ {}_5\theta_2^2 \leqq {}_5\overline{\Theta}_2^2 \ (= 3.5) \end{cases} \tag{43-2}$$

$$[g^{\theta}]\{{}_j\theta_i^3\} \leqq \{{}_j\overline{\Theta}_i^3\}, \text{ or } \begin{cases} {}_2\theta_1^3 \leqq {}_2\overline{\Theta}_1^3 \ (= 3.5) \\ {}_5\theta_2^3 \leqq {}_5\overline{\Theta}_2^3 \ (= 3.5) \end{cases} \tag{43-3}$$

Nonnegative Conditions

$$\phi_n^{\tau} \geqq 0 \ (\tau = 1 \sim 3,\ n = 1 \sim 6), \quad {}_j\theta_i^{\tau} \geqq 0 \ (\tau = 1 \sim 3,\ i = 1 \sim 2) \tag{44}$$

with

$$\underset{(6\times 6)}{[A]} = \begin{bmatrix} 1.04 & -0.48 & 0.00 & -0.48 & 0.00 & 0.00 \\ -0.48 & 2.16 & -0.48 & 0.04 & -0.96 & 0.04 \\ 0.00 & -0.48 & 1.04 & 0.00 & 0.00 & -0.48 \\ -0.48 & 0.04 & 0.00 & 1.08 & -0.48 & 0.00 \\ 0.00 & -0.96 & 0.00 & -0.48 & 2.08 & -0.48 \\ 0.00 & 0.04 & -0.48 & 0.00 & -0.48 & 1.08 \end{bmatrix} \tag{45}$$

$$\underset{(6\times 6)}{[\tfrac{1}{\Delta t}\,C]} = \begin{bmatrix} 0.06 & 0.03 & 0.00 & 0.03 & 0.00 & 0.00 \\ 0.03 & 0.23 & 0.03 & 0.06 & 0.06 & 0.06 \\ 0.00 & 0.03 & 0.06 & 0.00 & 0.00 & 0.03 \\ 0.03 & 0.06 & 0.00 & 0.12 & 0.03 & 0.00 \\ 0.00 & 0.06 & 0.00 & 0.03 & 0.12 & 0.03 \\ 0.00 & 0.06 & 0.03 & 0.00 & 0.03 & 0.12 \end{bmatrix} \tag{46}$$

$$\underset{(6\times 2)}{[D]} = \begin{bmatrix} 0 & 0 \\ -1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -1 \\ 0 & 0 \end{bmatrix} \tag{47}$$

$$\underset{(2\times 6)}{[g^{\phi}]} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{48}$$

$$\underset{(2\times 2)}{[g^{\theta}]} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{49}$$

$$\begin{Bmatrix} \Phi_1^0 \\ \Phi_2^0 \\ \Phi_3^0 \\ \Phi_4^0 \\ \Phi_5^0 \\ \Phi_6^0 \end{Bmatrix} = \begin{Bmatrix} 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \end{Bmatrix} \tag{50}$$

$$
\begin{Bmatrix} Q_1^1 \\ Q_2^1 \\ Q_3^1 \\ Q_4^1 \\ Q_5^1 \\ Q_6^1 \end{Bmatrix} = \begin{Bmatrix} Q_1^2 \\ Q_2^2 \\ Q_3^2 \\ Q_4^2 \\ Q_5^2 \\ Q_6^2 \end{Bmatrix} = \begin{Bmatrix} Q_1^3 \\ Q_2^3 \\ Q_3^3 \\ Q_4^3 \\ Q_5^3 \\ Q_6^3 \end{Bmatrix} = \begin{Bmatrix} 2.0 \\ 0.0 \\ 2.0 \\ 0.0 \\ 0.0 \\ 0.0 \end{Bmatrix} \tag{51}
$$

in which $\Delta t = 3600$ sec.

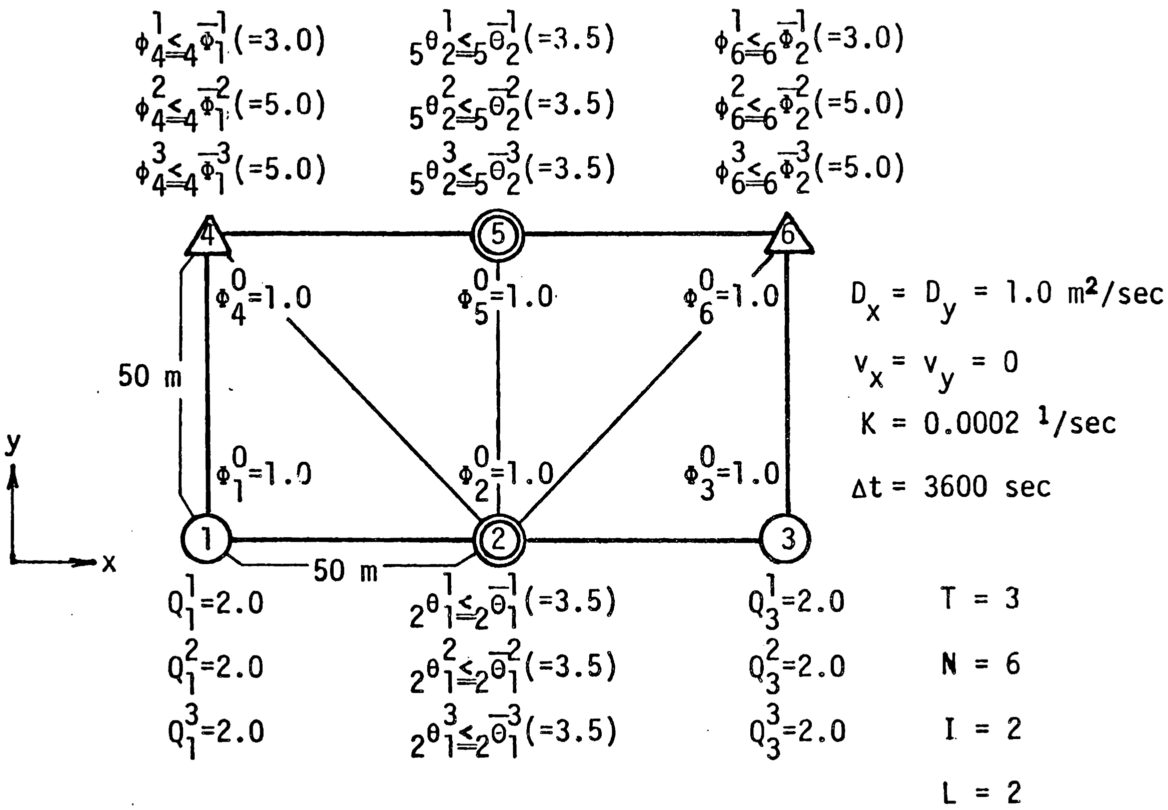
The computed results for the controllable loads (the decision variables) and the water qualities (the state variables) at each time step are shown in Figs. 5-4(a), 5-4(b) and 5-4(c).

It should be noted that the distribution patterns of the water qualities in Figs. 5-4(a), 5-4(b) and 5-4(c) arise from the resultant loads composed of the given uncontrollable loads $\{\{Q_n^\tau\}\}$ and the obtained controllable loads $\{\{{}_j\theta_i^\tau\}\}$.

The objective function is as follows:

$$
\begin{aligned}
Z &= \Delta t \times ({}_2\theta_1^1 + {}_5\theta_2^1 + {}_2\theta_1^2 + {}_5\theta_2^2 + {}_2\theta_1^3 + {}_5\theta_2^3) \\
&= 3600 \times (3.5 + 0.5 + 2.1 + 0.0 + 3.5 + 0.0) = 3\text{[illegible]}00 \qquad (52)
\end{aligned}
$$

By extending the efficient computational algorithm of FELP Method mentioned in Chapter 4, we can obtain an efficient computational algorithm of T. FELP Method. All of the computations in this chapter were performed by using slide rule.



$\partial\phi/\partial n = 0$ (on the four boundaries)

◎: Nodal point fitted for outfall of controllable load

○: Nodal point fitted for outfall of uncontrollable load

△: Regulated nodal point in water quality requirement

Fig. 5-3. Input Data on T. FELP Method in a Simple Model Basin

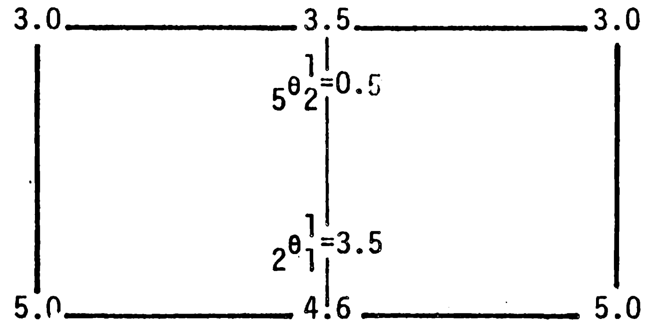


Fig. 5-4(a). Water Quality Distribution and Controllable Loads (τ = 1)

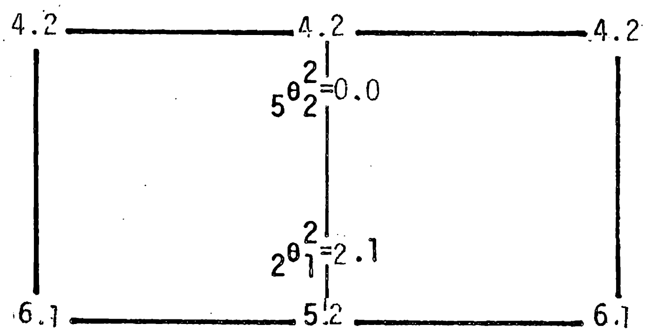


Fig. 5-4(b). Water Quality Distribution and Controllable Loads (τ = 2)

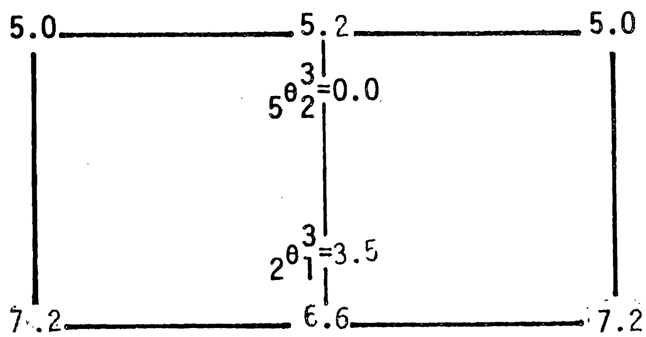


Fig. 5-4(c). Water Quality Distribution and Controllable Loads (τ = 3)

5-5. Concluding Remarks

A transient finite element & linear programming method (T. FELP Method) was described through the application to the control of unsteady state diffusion-convection phenomena in water pollution problems. The tractability not only in the initial and boundary conditions but also in the equality or inequality constraints makes sure that T. FELP Method becomes one of the useful techniques for several new types of boundary value problems.

Most practical applications of linear programming make use of the degital computer and existing computer codes based on two phases of the simplex method. However, in order to save computer time and memory, several efficient computational algorithms of T. FELP Method could be developed by taking note of the fact that the method has special structures.

The applicability of T. FELP Method is shown through a numerical example of water pollution problem governed by diffusion equation. The method makes it possible to obtain not only the optimal discharges from the various types of outfall to meet water quality requirements, but also the distribution patterns of several water qualities in the water basin simultaneously. A new criterion for selecting the locations of outfalls and the optimal volumes of discharged waste water may be given by T. FELP Method. The method may become an useful technique for analysis, planning and assessment in environmental and water resources problems.

In a manner similar to T. FELP Method, transient finite difference & linear programming Method (T. FDLP Method) could be developed by the combined use of finite difference method with linear programming. Aguado and Remson made a pioneering research associated with T. FDLP Method in the field of ground water management (1).

The related methods such as transient finite element & non-linear programming method and transient finite element & integer programming method could be developed. The developments certainly make it possible to solve more complicated and practical problems.

References

1. Aguado, E. and Remson, I., "Ground-Water Hydraulics in Aquifer Management," Journal of the Hydraulics Division, ASCE, Vol. 100, No. HY1, January, 1974, pp. 103-118.
2. Bellman, R., "Dynamic Programming," Princeton University Press, 1957.
3. Bruch, J.C., JR. and Zyvoloski, G., "Transient Two-Dimensional Heat Conduction Problems Solved by the Finite Element Method," International Journal for Numerical Methods in Engineering, Vol 8, 1974, pp. 481-497.
4. Comini, G., Guidice, D.S., Lewis, W.R. and Zienkiewicz, O.C., "Finite Element Solution of Non-Linear Heat Conduction Problems with Special Reference to Phase Change," International Journal for Numerical Methods in Engineering, Vol. 8, 1974, pp. 613-624.
5. Dantzig, B.G., "Linear Programming and Extensions," Princeton University Press, 1963.
6. Fan, L.T., "The Continuous Maximum Principle," John Wiley & Sons, New York, 1967.
7. Futagami, T., "Finite Element & Linear Programming Method and Water Pollution Control," Proceedings, 16th Congress of the International Association for Hydraulic Research, Vol. 3, C7, July-August, 1975, pp. 54-61.
8. Futagami, T., "Dynamic Programming for a Sewage Treatment System," Proceedings, 5th International Water Pollution Research Conference, July, 1970, Jenkins, H.S., ed., Pergamon Press Ltd., pp. II-21/1-21/12, 1971.
9. Gass, I.S., "Linear Programming," 3rd ed., McGraw-Hill Kogakusha, 1969.
10. Harleman, D.R.F., "Innovations in Heat Disposal in the Oceans," 2nd

Annual Sea Grant Lectures and Symposium, October, 1973, MIT Sea Grant Program Report, MIT, SG. 74-7, 1974.

11. Iwasa, Y. and Yatsuzuka, M., "Spread of Heated Waters from Multi-Port Diffuser," Proceedings, U.S.-Japan Seminar on Engineering and Environmental Aspects of Waste Heat Disposal, Printed at the Department of Civil Engineering, Kyoto University, Kyoto, 1974.

12. Koh, R.C.Y., Brooks, N.H., List, E.J. and Wolanski, J.E., "Thermal Outfall Diffusers for the San Onofre Nuclear Power Plant," Report No. KH-R30, W.M. Keck Laboratory of Hydraulics and Water Resources, C.I.T. California, 1974.

13. Pontryagin, L.S., Boltyanskii, R.V., Gamkrelidze, R.V., and Mischenko, E.F., "The Mathematical Theory of Optimal Processes," Wiley-Interscience, 1962, (English Translation by Trirogoff).

14. Smith, I.M., Farraday, R.V., and O'Connor, "Rayleigh-Ritz and Galerkin Finite Elements for Diffusion-Convection Problems," Water Resources Research, Vol. 9, No. 3, 1973, pp. 593-606.

15. Tamai, N., "Dispersion Models in Coastline Waters with Predominant Transverse Shear," Coastal Engineering in Japan, Japan Society of Civil Engineers, Vol. 17, 1974, pp. 185-197.

16. Tamai, N., Wiegel, L.R. and Tornberg, F.G., "Horizontal Surface Discharge of Warm Water Jets," Journal of Power Division, ASCE, Vol. 95, No. PO2, 1969, pp. 253-276.

17. Wada, A., "Study on Prediction Method of Simulation Analysis for Diffusion of Discharged Warm Water," Proceedings, U.S.-Japan Seminar on Engineering and Environmental Aspects of Waste Heat Disposal, Printed at the Department of Civil Engineering, Kyoto University, Kyoto, 1974.

18. Wilson, L.E., "The Determination of Temperatures within Mass Concrete Structures," Report No. 68-17, Structural Engineering Laboratory, University of California, Berkeley, California, 1968.

19. Zienkiewicz, O.C., "The Finite Element Method in Engineering Science, 2nd ed., McGraw-Hill, 1971.

Notations

The following symbols are used in Chapter 5.

$[A] = [a_{np}]$ = state matrix = global stiffness matrix, $(N \times N)$ matrix;

b = constant in governing equation;

$\{b_n^\tau\}$ = vector of constants in equilibrium equations at τth time step;

$\{\{b_l^g\}\}$ = vector of constants in constraints;

$[C] = [c_{np}]$ = capacity matrix, $(N \times N)$ matrix;

c_1, c_2 = coefficients in governing equation;

${}^{\theta}c_i^\tau$ = decision-evaluation coefficient = cost coefficient associated with ${}_j\theta_i^\tau$;

${}^{\phi}c_n^\tau$ = state-evaluation coefficient = cost coefficient associated with ϕ_n^τ;

$[D] = [d_{ni}]$ = decision matrix, $(N \times I)$ matrix;

D_{xk} = diffusion coefficient (D_x or D_y), L^2T^{-1};

$D.E.$ = differential equation (governing equation);

e = initial condition;

f = objective function;

$[G_\theta]$ = decision-constraint matrix, $(L_T \times (T \times I))$ matrix;

$[G_\phi]$ = state-constraint matrix, $(L_T \times (T \times N))$ matrix;

g = constraint;

$[g^\phi] = [g_{ln}^\phi]$ = sub-state-constraint matrix, $(L \times N)$ matrix;

$[g^\theta] = [g_{ii}^\theta]$ = sub-decision-constraint matrix, $(I \times I)$ unit matrix;

h = boundary condition;

I = number of decision variables at each time step (number of controllable loads, i.e., number of nodal points fitted for locations of outfalls of controllable loads);

$i = 1 \sim I$ = decision variable number at each time step (controllable load number at each time step);

j = nodal point number associated with ith decision variable (nodal point number fitted for location of ith controllable load);

K = decay factor (heat transfer coefficient at water surface or decay rate of pollutant), T^{-1};

k = index associated with Cartesian coordinates;

L = number of regulated nodal points in water quality requirements at each time step;

L_T = total number of constraints;

$l = 1 \sim L$ = water quality requirement number at each time step;

$l = 1 \sim L_T$ = constraint number in general T. FELP Method;

l_{xk} = direction cosine outward normal to boundary, (l_x or l_y);

m = regulated nodal point number in lth water quality requirement;

$[\mathrm{N}] = [N_1 \, .. \, N_n \, .. \, N_N]$ = shape functions (weighting functions);

N = number of state variables at each time step (total number of nodal points in finite elements);

$n, p = 1 \sim N$ = state variable number at each time step (nodal point number);

Q = uncontrollable load;

Q_n^τ = uncontrollable load at nodal point n at τth time step;

q = intensity of flux per unit length of boundary;

S = boundary ($S^1 + S^2$), L or L^2;

S^1, S^2 = parts of boundary, L or L^2;

s^2 = segment of part of boundary S^2, L or L^2;

T = total number of time steps;

t = time, T;

Δt = increment in time, T;

v_k = convective velocity (v_x or v_y), LT^{-1};

x_k = Cartesian coordinate (x, y or z), L;

X_k = Cartesian coordinate of boundary (X, Y or Z), L;

Z = objective function;

κ = decay factor of boundary, LT^{-1};

θ = decision variable (controllable load);

$\{\{\theta\}\}$ = $\{\{{}_j\theta_i^\tau\}\}$ = vector of decision variables in whole domain Ω (vector of controllable loads in whole domain Ω);

$\{{}_j\theta_i^\tau\}$ = vector of decision variables at τth time step;

$\underline{\Theta}$ = lower limit of decision variable;

$\overline{\Theta}$ = upper limit of decision variable (upper limit of controllable load);

${}_j\overline{\Theta}_i^\tau$ = upper limit of ith controllable load;

ϕ = state variable (water quality in water basin);

$\{\{\phi\}\}$ = $\{\{\phi_n^\tau\}\}$ = vector of state variables in whole domain Ω (vector of water qualities in whole domain Ω);

$\{\phi_n^\tau\}$ = vector of state variables at τth time step (vector of water qualities at τth time step);

$\{\Phi_n^0\}$ = vector of initial states (vector of initial water qualities);

Φ^0 = initial state (initial water quality);

$\underline{\Phi}$ = lower limit of state variable;

$\overline{\Phi}$ = upper limit of state variable (water quality requirement);

Φ_a = temperature or concentration of surrounding boundary;

Φ_b = prescribed boundary value;

Φ_b^τ = prescribed boundary value at τth time step;

${}_m\overline{\phi}_l^{\tau}$ = lth water quality requirement at τth time step;

$\{\frac{\partial\phi}{\partial t}|_n^{\tau}\}$ = vector of rate of change of state variables at τth time step;

$\tau = 1 \sim T$ = time step number;

$\Omega = \Omega_s \times \Omega_t$ = whole domain, L^2T or L^3T;

Ω_s = whole space domain (whole water basin), L^2 or L^3;

$\Omega_t = T \times \Delta t$ = whole time domain, T;

Ω^g = subdomain associated with constraints, L^2T or L^3T;

Ω_s^g = sub-space-domain associated with constraints, L^2 or L^3;

Ω_t^g = sub-time-domain associated with constraints, T;

Δ = area of each element, L^2;

Chapter 6

CONCLUSIONS

In this investigation three fundamental mathematical methods in water polluiton control were presented: A finite element & linear programming method (FELP Method, or, the F.E. & L.P. Method), a transient finite element & linear programming method (T. FELP Method), and a finite difference & linear programming method (FDLP Method, or, the F.D.& L.P. Method) were developed and systematized.

An efficient computational algorithm for the proposed methods was developed in order to reduce computer time and memory by noting the fact that the proposed methods have special structures.

An analytical method based on double Fourier series was also developed in order to check the computations of FELP Method and FDLP Method. The results of the analytical method showed good agreement with those of FELP Method and FDLP Method.

The applicability of the proposed methods was shown through numerical examples of water pollution problems governed by the convective-diffusion equation. The results of FDLP Method (see Table 2-2 and Figs. 2-6(a),(b)) were compared favourably with those of FELP Method (see Table 3-2 and Figs. 3-6(a), (b)) under the same computational conditions.

It seems that the proposed methods give us new criteria for selecting the locations of outfalls and the optimal volumes of discharged waste water.

The proposed methods may become fundamental methods for a systems approach in environmental and water resources problems. The proposed methods may become useful techniques for analysis, planning, assessment and management in environmental and water resources problems. Especially, FELP Method and T. FELP Method are useful because of their generality with respect to geometry, material properties and boundary conditions.

Tractability not only in the initial and boundary conditions but also in the equality or inequality constraints makes sure that the proposed methods become powerful techniques for several new types of boundary value problems in various fields.

The development of an efficient computational algorithm for the proposed methods makes it possible to attack large scale problems. In FELP Method and T. FELP Method skillful techniques based on the band matrix of the finite element method could be used in Gaussian Elimination in order to reduce futher computation time and memory.

Finally, future aspects associated with the proposed methods should be mentioned.

Related methods such as a finite element (difference) & non-linear programming method, a finite element (difference) & integer programming method, and a stochastic finite element (difference) & linear programming method could be developed. Such developments would probably make it possible to solve more complicated and realistic problems. Such methods may make it possible to attack complex coupling problems. One such problems is the coupling problem governed by the equations of motion and continuity and the convective-diffusion equation.

Comparison with other analytical methods, experiments and field data should be extended to make the applicability of the proposed methods wider.

[redacted]

長岡先生の休学 ——一九六七年——
湯川秀樹自選集 第四巻 創造の世界
朝日新聞社

APPENDICES. COMPUTER PROGRAMS

A. Computer Program for Finite Element & Linear Programming Method - Triangular Element with Six Nodal Points

SOURCE STATEMENT

```
C     MAIN      FLP26    ONLY
      COMMON/DOM    /IMOJI(20)
      COMMON/DATAOR/NF261,NF262,NF263,IEK,INK,IFK,IFD,KNI
      COMMON/FEMFD/JFEMFD
      JFEMFD=0
      IFD=2
      INK=3
      IEK=3
      KNI=6
      READ(5,520) ICASE
  520 FORMAT(I5)
      WRITE(6,620)ICASE
  620 FORMAT(1H1,//20X,'ICASE =',I5,//)
      DO 10 NCASE=1,ICASE
      WRITE(6,622)NCASE
  622 FORMAT(1H1,//,20X,'*** NCASE =',I5,' ***',10(/))
      READ(5,505)IMOJI
  505 FORMAT(20A4)
      WRITE(6,650)IMOJI
  650 FORMAT(1H0,5(/),19X,32(3H*  ),//19X,1H*,92X,1H*,//19X,1H*,5X,
     1       20A4,7X,1H*,//19X,1H*,92X,1H*,//19X,32(3H*  ),5(/))
      READ(5,521)IFK
  521 FORMAT(5X,I5)
      WRITE(6,621)IFK
  621 FORMAT(1H1,2X,' * 0 *',//3X,'IFK =',I5,/)
      WRITE(6,6113)
 6113 FORMAT(1H0,30X,'FLP26',//)
      CALL FLP26
   10 CONTINUE
      STOP
      END
```

SOURCE STATEMENT

```
C      SUBROUTINE
        SUBROUTINE FLP26
        COMMON/SLP1  /A(345,355),B(500),CC(500)
        COMMON/SLP2  /M1,M2,M3,N0,NAME(5),JLPW,NITERE
        COMMON/SLP3  /BP(500),C(500),IB(500),NUM(500)
C       COMMON/SLP4  /Z,ZERO,ONE,PGRE,EPS
        COMMON/SLP4  /ZOB,ZERO,ONE,PGRE,EPS
        COMMON/SLP5  /M,N,NC,N1F(11),N2F(5),LS,NAX,MAX
        COMMON/SLP6  /CB(450),CCC(450)
        COMMON/SLP7  /XL(200),NXLS(200),NXLE(200),NFUS(200),NFUE(200)
        COMMON/SLP8  /LBXL(200),LBXLE,KFU(200),KFUE
        COMMON/SLP9  /MXLG,NFUG,NFUN,LPDT
        COMMON/SLP10 /AIKA(50),IIKA(50),JIKA(50),IKA
        COMMON/SLP11 /BIKB(50),CCIKCC(50),IIKB(50),IIKCC(50),IKB,IKCC
        COMMON/SLP12 /INITIA
        COMMON/SLP14 /KPHASE
        COMMON/SLP15 /QCL(200),LQ(200),LFUE,LQE,LQQ(200),LIQ(200)
        COMMON/SJEXP/JEXP
        COMMON/SPF26 /IPR
        COMMON/ACHECK/IACH
        COMMON/SPA   /MACP,MICP
        COMMON/SAUT/  XSTART,YSTART,XEND,YEND,DMX,DMY,MXE,MYE
        COMMON/DATAHF/IHF
        COMMON/FKIND/ KIND,KINDD,KINDK,KINDV
        COMMON/SPART/ICN(20,3),ICNS(20,3),ICNE(20,3),IPTE,KEPE(20)
        COMMON/DATA0R/NF331,NF332,NF333,IEK,INK,IFK,IFD,KNI
        COMMON/DATA1R/IAD,KFEM1,MTITL(5)
        COMMON/DATA2R/Z(500,2),NP
        COMMON/DATA3R/AREAS,SEIDO,NE,INKE,IAREA,ISEIDO,NOD(500,6)
        COMMON/DATA3C/CSEIDO
        COMMON/DATA6R/ME,JQD
        COMMON/DATA7R/VE(500,2),VN(500,2),IVEL,IVG,IVEX,IVW,IEXP
        COMMON/DATA8R/BV(300,1),BL(300),NBN,JBN,NF(300)
        COMMON/DATA8C/NNFF(500)
        COMMON/DATA9R/KEISAN,IAB,NAB(500)
        COMMON/DATA68/BC(500,1)
        COMMON/SJAXB /JAXB
        COMMON/SSKMT/HH(6,6)
        COMMON/RESULT/X(500)
C       COMMON/SUNIB /B(500,1)
        COMMON/SUNIB /B0(500,1)
        COMMON/SBCHEK/IBCHEK(20) ,IBCH
        COMMON/DOM    /IMOJI(20)
        COMMON/D12/JFIG
        COMMON/ACRACA/ACR(20,50),ACA(5,120)
        LOGICAL CB
        WRITE(6,6161)
  6161  FORMAT(1H0,10X,'FINITE ELEMENT & LINEAR PROGRAMMING METHOD',//)
        ME=1
        IFK=4
        NAX=450
        MAX=300
        CALL DATA26
        DO 386 I=1,N0
   386  X(I)=0.0
        IF(INITIA.NE.0) N=N0+M1+M2
```

SOURCE STATEMENT

```
      IF(ISEIDO.NE.0.AND.SEIDO.LE.CSEIDO) GO TO 80
      IF(KEISAN.EQ.0) GO TO 82
      GO TO 83
   80 WRITE(6,610)
  610 FORMAT(1H0,5X,'** KEISAN SINAI (SEIDO.LE.CSEIDO) **',/)
      GO TO 130
   82 WRITE(6,612)
  612 FORMAT(1H0,5X,'** KEISAN SINAI (KEISAN.EQ.0) **',/)
      GO TO 130
   83 CONTINUE
CCCCCCCCC
      ZERO =0.0E0
      DO 50 I=1,M
      DO 51 J=1,N
   51 A(I,J)=ZERO
   50 CONTINUE
      DO 52 I=1,M
   52 B(I)=ZERO
      DO 53 J=1,N
   53 CC(J)=ZERO
      KM1=0
      DO 122 I=1,MXLG
      KS=NXLS(I)
      KE=NXLE(I)
      DO 123 KK=KS,KE
      KM1=KM1+1
      LBXL(KM1)=KK
      B(KM1)=XL(I)
  123 A(KM1,KK)=1.0
  122 CONTINUE
      LBXLE=KM1
      KFUE=LFUE
      IF(LFUE.EQ.0) GO TO 90
      DO 91 I=1,LQE
      LI=LBXLE+I
      NLQ=NP+LIQ(I)
      LIQL=LIQ(I)
      B(LI)=QCL(LIQL)
   91 A(LI,NLQ)=1.0
      DO 92 I=1,LFUE
      KK=KFU(I)
      NPI=NP+I
      A(KK,NPI)=-1.0
      CC(NPI)=1.0
   92 CONTINUE
      GO TO 93
   90 CONTINUE
      KF1=0
      DO 260 I=1,NFUG
      KS=NFUS(I)
      KE=NFUE(I)
      DO 260 KK=KS,KE
      KF1=KF1+1
      KFU(KF1)=KK
      KF2=KF1+NP
      MP=M1+KK
```

SOURCE STATEMENT

```
      A(MP,KF2)=-1,0
      CC(KF2)=1.0
      IF(IAB.EQ.0) GO TO 260
      WRITE(6,686)I,KK,KF1,KFU(KF1),KF2,MP,A(MP,KF2),CC(KF2)
  686 FORMAT(1H ,5X,'I,KK,KF1,KFU(KF1),KF2,MP,A(MP,KF2),CC(KF2)',5X,
     1       6I5,2E12.4)
  260 CONTINUE
      KFUE=KF1
      WRITE(6,685)LBXLE,KFUE
  685 FORMAT(1H0,5X,'LBXLE =',I5,5X,'KFUE =',I5,/)
   93 CONTINUE
CCCCCCCCCC
      IF(INITIA.NE.0) GO TO 8000
      DO 723 I=1,NP
      DO 723 MM=1,ME
  723 BC(I,MM)=BO(I,MM)
      IF(NBN.EQ.0) GO TO 809
      DO 801 I=1,NP
  801 NNFF(I)=0
      DO 802 K=1,NBN
      KF=NF(K)
      NNFF(KF)=1
      DO 803 MM=1,ME
  803 BO(KF,MM)=BV(K,MM)
  802 CONTINUE
  809 CONTINUE
      DO 200 KM=1,NE
      CALL SKMT(KM)
      DO 200 KI=1,KNI
      MI=NOD(KM,KI)
      MI=MI+M1
      DO 200 KJ=1,KNI
      MJ=NOD(KM,KJ)
  200 A(MI,MJ)=A(MI,MJ)+HH(KI,KJ)
      IF(IKA.EQ.0) GO TO 201
      DO 202 I=1,IKA
      KA=IIKA(I)
      JKA=JIKA(I)
  202 A(KA,JKA)=AIKA(I)
  201 CONTINUE
CC BCHECKS
      IF(IBCH.EQ.0) GO TO 292
      DO 238 I=1,IBCH
      MP=IBCHEK(I)+M1
      DO 238 J=1,NP
  238 ACA(I,J)=A(MP,J)
CC    BCHECKE
  292 CONTINUE
      IF(NBN.EQ.0) GO TO 211
      DO 210 I=1,NBN
      KF=NF(I)
      MKF=KF+M1
CC    BRCHECKS
      IF(KEISAN.LT.2) GO TO 71
      DO 70 J=1,NP
   70 ACR(I,J)=A(MKF,J)
```

SOURCE STATEMENT

```
   71 CONTINUE
 CC    BRCHEKE
      DO 210 J=1,NO
      A(MKF,J)=0.0
      IF(KF.EQ.J) A(MKF,J)=1.0
  210 CONTINUE
  211 CONTINUE
      DO 300 I=1,NP
      MP=M1+I
  300 B(MP)=BO(I,1)
      IF(IKB.EQ.0) GO TO 203
      DO 204 I=1,IKB
      KB=IIKB(I)
  204 B(KB)=BIKB(I)
  203 CONTINUE
 8000 CONTINUE
      WRITE(6,640)
  640 FORMAT(1H1,//,15X,'B(I)',//)
      DO 920 I=1,LBXLE
      IK=LBXL(I)
      WRITE(6,641)I,IK,B(I)
  641 FORMAT(1H ,5X,2I5,E12.4)
  920 CONTINUE
      IF(LFUE.EQ.0) GO TO 95
      I1=LBXLE+1
      DO 94 I=I1,M1
      IL=I-LBXLE
      IK=LQQ(IL)
      WRITE(6,641)I,IK,B(I)
   94 CONTINUE
   95 CONTINUE
      DO 921 I=1,NP
      II=I+M1
      WRITE(6,691)II,I,B(II)
  691 FORMAT(1H ,5X,2I5,E12.4)
  921 CONTINUE
      WRITE(6,642)
  642 FORMAT(1H1,//,15X,'CC(I)',//)
      DO 922 I=1,NP
      WRITE(6,643)I,CC(I)
  643 FORMAT(1H ,5X,I5,5X,E12.4)
  922 CONTINUE
      DO 923 I=1,KFUE
      II=I+NP
      IK=KFU(I)
      WRITE(6,692)II,IK,CC(II)
  692 FORMAT(1H ,5X,2I5,E12.4)
  923 CONTINUE
      IF(INITIA.NE.0) GO TO 8010
      IF(IACH.LT.1) GO TO 925
      DO 924 I=1,M
      IF(I.GT.M1) GO TO 926
      IF(I.GT.LBXLE) GO TU 96
      IK=LBXL(I)
      WRITE(6,693)I,IK
  693 FORMAT(1H0,5X,2I5)
```

SOURCE STATEMENT

```
      GO TO 927
   96 IL=I-LBXLE
      IK=LQQ(IL)
      WRITE(6,693) I,IK
      GO TO 927
  926 II=I-M1
      WRITE(6,694)I,II
  694 FORMAT(1H0,5X,2I5)
  927 WRITE(6,695)(A(I,J),J=1,N0)
  695 FORMAT(1H ,/(5X,10E12.4))
  924 CONTINUE
  925 CONTINUE
 8010 CONTINUE
      CALL LIPR
      IF(KPHASE.EQ.1) GO TO 130
      WRITE(6,670)
  670 FORMAT(1H1,//,20X,' LP RESULT',//)
      WRITE(6,688)ZOB
      DO 295 I=1,M
      II=IB(I)
  295 X(II)=B(I)
      WRITE(6,620)(I,X(I),I=1,M)
  620 FORMAT(1H ,/(5X,I5,E12.4))
      WRITE(6,621)
  621 FORMAT(1H1,//,20X,'RESULT',//)
      WRITE(6,688)ZOB
  688 FORMAT(1H0,//,10X,  'OBJECT FUNCTION =',E16.7,//)
      DO 296 I=1,NP
      WRITE(6,622)(I,(Z(I,J),J=1,IFD),X(I))
  622 FORMAT(1H ,5X,I5,5X,2F10.2,E12.4)
  296 CONTINUE
      WRITE(6,624)
  624 FORMAT(1H0,/)
      DO 297 IK=1,KFUE
      II=KFU(IK)
      I=NP+IK
      WRITE(6,623)(I,II,(Z(II,J),J=1,IFD),X(I))
  623 FORMAT(1H ,5X,2I5,2F10.2,E12.4)
  297 CONTINUE
CCC   FIGURE START
      IF(JFIG.EQ.0) GO TO 1283
CHUI S.AUT.NO KXE.KYE DEWA NAI
      IF(KNI.EQ.3) GO TO 9025
      KXE=MXE*2+1
      KYE=MYE*2+1
      GO TO 9026
 9025 CONTINUE
      KXE=MXE+1
      KYE=MYE+1
 9026 CONTINUE
      WRITE(6,650)IMOJI
  650 FORMAT(1H1,5(/),19X,32(3H*  ),//19X,1H*,92X,1H*,//19X,1H*,5X,
     1      20A4,7X,1H*,//19X,1H*,92X,1H*,//19X,32(3H*  ),5(/))
      WRITE(6,6161)
      IF(KXE.GT.21) GO TO 1288
      WRITE(6,6660)
```

SOURCE STATEMENT

```
      DO 1981 I=1,KYE
      IF(JFIG)1982,1983,1984
 1982 NS=(I-1)*KXE+1
      NH=I*KXE
      IY=I
       GO TO 1985
 1984 NS=NP-I*KXE+1
      NH=NP-(I-1)*KXE
      IY=KYE-I+1
 1985 WRITE(6,6961)IY,(IN,IN=NS,NH)
 6961 FORMAT(1H0,//,I3,1X,21I6)
 1981 CONTINUE
      WRITE(6,6962)(IX,IX=1,KXE)
 6962 FORMAT(1H0,///,4X,21I6)
 1983 CONTINUE
      DO 1290 MM=1,ME
      WRITE(6,6660)
 6660 FORMAT(1H1,//)
      WRITE(6,6963)IMOJI
 6963 FORMAT(1H ,25X,20A4,//)
      DO 1281 I=1,KYE
       IF(JFIG) 1282,1283,1284
 1282 NS=(I-1)*KXE+1
      NH=I*KXE
      IY=I
      GO TO 1285
 1284 NS=NP-I*KXE+1
      NH=NP-(I-1)*KXE
      IY=KYE-I+1
 1285 WRITE(6,6661)IY,(X (IN    ),IN=NS,NH)
 6661 FORMAT(1H0,//,I3,1X,21F6.1)
 1281 CONTINUE
      WRITE(6,6962)(IX,IX=1,KXE)
 1290  CONTINUE
      GO TO 1283
 1288 WRITE(6,6662)
 6662 FORMAT(1H0,//,20X,'KXE.GT.21 DE FIGURE NASI',/)
 1283 CONTINUE
CCC   FIGURE END
      DO 724 I=1,NP
      DO 724 MM=1,ME
  724 BO(I,MM)=BC(I,MM)
CC    BCHECKS
      IF(IBCH.EQ.0) GO TO 400
      WRITE(6,690)
  690 FORMAT(1H1,//,20X,'LOAD CHECK',//)
      DO 299 I=1,IBCH
      IBCK=IBCHEK(I)
      II=IBCK+M1
      CEB=0.0
      DO 298 J=1,NP
  298 CEB=CEB+ACA(I,J)*X(J)
      WRITE(6,630)II,I,IBCHEK(I),(Z(IBCK,K),K=1,IFD),CEB,BO(IBCK,1)
  630 FORMAT(1H ,5X,3I5,2F10.2,2E12.4)
  299 CONTINUE
  400 CONTINUE
```

SOURCE STATEMENT

```
CC      BCHECKE
CC      BRCHEKS
        IF(KEISAN.LT.2) GO TO 75
        IF(NBN.EQ.0) GO TO 75
        WRITE(6,631)
   631  FORMAT(1H1,//,20X,'REACTION LOAD',//)
        DO 76 I=1,NBN
        KF=NF(I)
        MKF=KF+M1
        BCRB=0.0
        DO 77 J=1,NP
    77  BCRB=BCRB+ACR(I,J)*X(J)
        WRITE(6,632)MKF,I,KF,(Z(KF,K),K=1,IFD),BCRB
   632  FORMAT(1H ,5X,3I5,2F10.0,E12.4)
    76  CONTINUE
    75  CONTINUE
CC      BRCHEKE
        IF(IPR.EQ.0) GO TO 130
        CALL PF26
   130  CONTINUE
        RETURN
        END
```

SOURCE STATEMENT

```
C     SUBROUTINE
      SUBROUTINE DATA26
      COMMON/D12/JFIG
      COMMON/SAUT/  XSTART,YSTART,XEND,YEND,DMX,DMY,MXE,MYE
      COMMON/DATAHF/IHF
      COMMON/SJEXP/JEXP
      COMMON/SPF26 /IPR
      COMMON/ACHECK/IACH
      COMMON/SPA    /MACP,MICP
      COMMON/SUNIB /B(500,1)
      COMMON/SPART/ICN(20,3),ICNS(20,3),ICNE(20,3),IPTE,KEPE(20)
      COMMON/DATA0R/NF331,NF332,NF333,IEK,INK,IFK,IFD,KNI
      COMMON/DATA1R/IAD,KFEM1,MTITL(5)
      COMMON/DATA2R/Z(500,2),NP
      COMMON/DATA2X/ZP(100,2),NZP(100),NNPP
      COMMON/DATA3R/AREAS,SEIDO,NE,INKE,IAREA,ISEIDO,NOD(500,6)
      COMMON/DATA3C/CSEIDO
      COMMON/DATA4 /BU(50,8)
      COMMON/DATA5R/IS,ISCE
      COMMON/DATA5C/KBUK(500)
      COMMON/DATA6R/ME,JQD
      COMMON/DATA7R/VE(500,2),VN(500,2),IVEL,IVG,IVEX,IVW,IEXP
      COMMON/DATA8R/BV(300,1),BL(300),NBN,JBN,NF(300)
      COMMON/DATA8B/FLUX(300,5),CLOSS(300),ATZERO(300),TZERO(300)
     1       ,NBF(300),NBL(300,3),NBCE,NBCL,JCAT
      COMMON/DATA8C/NNFF(500)
      COMMON/DATA9R/KEISAN,IAB,NAB(500)
      COMMON/DATA68/BC(500,1)
      COMMON/SJAXB /JAXB
      COMMON/FKIND/ KIND,KINDD,KINDK,KINDV
      COMMON/SABCDX/DELTA( 500)
      COMMON/SBCHEK/IBCHEK(20) ,IBCH
      DIMENSION ISS(50),ISE(50),IKBU(50)
      DIMENSION VNG(300,3),IVNGS(300),IVNGE(300)
      DIMENSION BCQ( 300),NC( 300)
      DIMENSION NEAB(50)
      DIMENSION MOJI( 5)
      DIMENSION QQ(50),NDS(50),NDE(50)
      WRITE(6,600)
  600 FORMAT(1H0,44X,14H* INPUT DATA *,10X,//)
  100 READ(5,500)ID,ID1,ID2,ID3,ID4,ID5,RD1,RD2,MOJI
  500 FORMAT(I2,3X,5I5,2F15.0, 5A4)
      IF(ID.LT.1.OR.ID.GT.13) GO TO 110
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13),ID
    1 WRITE(6,601)ID,(MOJI(I),I=1, 5)
  601 FORMAT(1H0,//,2X,' *',I2,' *',5X,5A4)
      KIND=ID1
      KINDD=ID2
      KINDK=ID3
      KINDV=ID4
      IHF=ID5
      WRITE(6,610)KIND,KINDD,KINDK,KINDV,IHF
  610 FORMAT(1H0,3X,'KIND =',I3,5X,'KINDD =',I3,5X,
     1       'KINDK =',I3,5X,'KINDV =',I3,5X,'IHF =',I3/)
      DO 15 I=1,5
      MTITL(I)=MOJI(I)
```

SOURCE STATEMENT

```
   15 CONTINUE
      GO TO 100
    2 WRITE(6,602)ID,(MOJI(I),I=1,5)
  602 FORMAT(1H0,//,2X,' *',I2,' *',5X,5A4)
      IAD=ID1
      NP=ID2
      NNPP=ID3
      WRITE(6,620)IAD,NP,NNPP
  620 FORMAT(1H0,3X,'IAD =',I3,5X,'NP =',I4,5X,'NNPP =',I4,/)
      IF(IAD.EQ.0) GO TO 20
      CALL AUT
      GO TO 100
   20 IF(KNI.EQ.3) GO TO 22
      READ(5,520)(NZP(I),(ZP(I,J),J=1,IFD),I=1,NNPP)
  520 FORMAT(3(I5,2F10.0))
      WRITE(6,621)(I,NZP(I),(ZP(I,J),J=1,IFD),I=1,NNPP)
  621 FORMAT(1H ,/(3(3X,2I5,1X,2F10.2)))
      DO 21 I=1,NNPP
      II=NZP(I)
      DO 21 J=1,IFD
      Z(II,J)=ZP(I,J)
   21 CONTINUE
      GO TO 100
   22 READ(5,5200)(Z(I,1),Z(I,2),I=1,NP)
 5200 FORMAT(4(2F10.0))
      WRITE(6,6210)(I,Z(I,1),Z(I,2),I=1,NP)
 6210 FORMAT(1H ,/(4(3X,I4,1X,2F10.4)))
      GO TO 100
    3 WRITE(6,601)ID,(MOJI(I),I=1, 5)
      IF(IAD.EQ.0) NE=ID1
      INKE=ID2
      IAREA=ID3
      ISEIDO=ID4
      AREAS=RD1
      SEIDO=RD2
      WRITE(6,630)NE,INKE,IAREA,ISEIDO,AREAS,SEIDO
  630 FORMAT(1H0,3X,4HNE =,I4,5X,6HINKE =,I3,5X,7HIAREA =,I3,5X,
     1  8HISEIDO =,I3,5X,7HAREAS =,E12.5,5X,7HSEIDO =,E12.5,/)
      IF(IAD.EQ.0) GO TO 30
      GO TO 31
   30 IF(KNI.EQ.3) GO TO 37
      READ(5,530)((NOD(I,J),J=1,KNI),I=1,NE)
  530 FORMAT(2(6I5))
      WRITE(6,631)(I,(NOD(I,J),J=1,KNI),I=1,NE)
  631 FORMAT(1H ,/(2(3X,I4,1X,6I5)))
   31 CONTINUE
      IF(KNI.EQ.3) GO TO 38
      DO 300 I=1,NE
      I1=NOD(I,1)
      I2=NOD(I,2)
      I3=NOD(I,3)
      I4=NOD(I,4)
      I5=NOD(I,5)
      I6=NOD(I,6)
      DO 300 K=1,IFD
      Z(I4,K)=(Z(I2,K)+Z(I3,K))/2.0
```

SOURCE STATEMENT

```
      Z(I5,K)=(Z(I3,K)+Z(I1,K))/2.0
  300 Z(I6,K)=(Z(I1,K)+Z(I2,K))/2.0
      GO TO 38
   37 READ(5,5300)((NOD(I,J),J=1,3),I=1,NE)
 5300 FORMAT(5(3I5))
      WRITE(6,6310)(I,(NOD(I,J),J=1,3),I=1,NE)
 6310 FORMAT(1H ,/(5(3X,I4,1X,3I5)))
   38 CONTINUE
      SAREA=0.0
      DO 35 I=1,NE
      CALL ABCDXY(I)
      SAREA=SAREA+DELTA(I)
   35 CONTINUE
      WRITE(6,632)SAREA
  632 FORMAT(1H0,5X,7HSAREA =,E12.5,/)
      IF(IAREA.GE.0) GO TO 36
      WRITE(6,633)(I,DELTA(I),I=1,NE)
  633 FORMAT(1H ,/(5X,5(I4,F12.4,3X)))
   36 CONTINUE
      IF(ISEIDO.EQ.0) GO TO 34
      CSEIDO=ABS((AREAS-SAREA)/SAREA)
      WRITE(6,634)CSEIDO,CSEIDO
  634 FORMAT(1H0, 5X,8HCSEIDO =,F10.6,10X,E12.5)
      IF(SEIDO.GT.CSEIDO) GO TO 34
      WRITE(6,635)
  635 FORMAT(1H0,5X,19HMENSEKI SEIDO WARUI,/)
      IF(ISEIDO.GT .0) GO TO 34
      STOP
   34 IF(INKE.EQ.0) GO TO 100
      CALL NKE
      GO TO 100
    4 WRITE(6,601)ID,(MOJI(I),I=1, 5)
      KBU=ID1
      WRITE(6,640)KBU
  640 FORMAT(1H0,3X,5HKBU =,I4,/)
      READ(5,540)((BU(I,IBU),IBU=1,8),I=1,KBU)
  540 FORMAT(8F10.0)
      WRITE(6,641)( I,(BU(I,IBU),IBU=1,8),I=1,KBU)
  641 FORMAT(1H ,/(3X,I4,8F12.4))
      GO TO 100
    5 WRITE(6,602)ID,(MOJI(I),I=1, 5)
      IS=ID1
      IS46=ID2
      ISCE=ID3
      WRITE(6,650)IS,IS46,ISCE
  650 FORMAT(1H0,3X,'IS =',I4,5X,'IS46 =',I3,5X,'ISCE =',I3,/)
      READ(5,550)(ISS(I),ISE(I),IKBU(I),I=1,IS)
  550 FORMAT(5(3I5))
      WRITE(6,651)(I,ISS(I),ISE(I),IKBU(I),I=1,IS)
  651 FORMAT(1H ,/(5(3X,I4,1X,3I5)))
      DO 50 I=1,IS
      KSS=ISS(I)
      KSE=ISE(I)
      IF(IS46.EQ.0) GO TO 52
      IF(IFD.EQ.3.AND.INK.EQ.4) KSS=(ISS(I)-1)*5+1
      IF(IFD.EQ.3.AND.INK.EQ.4) KSE=ISE(I)*5
```

SOURCE STATEMENT

```
   52 CONTINUE
      NEAM=IKBU(I)
      DO 50 M=KSS,KSE
      KBUK(M)=NEAM
   50 CONTINUE
      IF(ISCE.EQ.0) GO TO 100
      WRITE(6,653)
  653 FORMAT(1H0,/)
      WRITE(6,652)(I,KBUK(I),I=1,NE)
  652 FORMAT(1H ,/(5X,10(2I5)))
      GO TO 100
    6 WRITE(6,601)ID,(MOJI(I),I=1, 5)
      IF(IFK.EQ.3) GO TO 100
      ME=ID1
      JQD=ID2
      JQC=ID3
      IQW=ID4
      WRITE(6,662)ME,JQD,JQC,IQW
  662 FORMAT(1H0,3X,'ME =',I5,5X,'JQD =',I5,
     1       5X,'JQC =',I5,5X,'IQW =',I5/)
      DO  1670  I=1,NP
      DO 1670 MM=1,ME
 1670 B(I,MM)=0.0
      IF(JQD.EQ.0) GO TO 6690
      DO 6691 MM=1,ME
      IF(JQD.EQ.1.AND.MM.GE.2) GO TO 6694
      READ(5,5666)NDQ
 5666 FORMAT(I5)
      WRITE(6,5667)NDQ
 5667 FORMAT(1H0,5X,'NDQ =',I5,/)
      IF(NDQ.EQ.0) GO TO 6691
      READ(5,5668)(NDS(I),NDE(I),QQ(I),I=1,NDQ)
 5668 FORMAT(4(2I5,F10.4))
      WRITE(6,6664)(I,NDS(I),NDE(I),QQ(I),I=1,NDQ)
 6664 FORMAT(1H ,/(4(3X,3I4,F12.4)))
      DO 6692 I=1,NDQ
      KS=NDS(I)
      KE=NDE(I)
      DO 6692 J=KS,KE
      DO 1661 IJ=1,KNI
      II=NOD(J,IJ)
 1661 B(II,MM)=B(II,MM)+QQ(I)*DELTA(J)/FLOAT(KNI)
 6692 CONTINUE
      GO TO 6691
 6694 DO 6696 K=1,NP
 6696 B(K,MM)=B(K,1)
 6691 CONTINUE
 6690 CONTINUE
      IF(JQC.EQ.0) GO TO 1609
      DO 1660 MM=1,ME
      IF(JQC.EQ.1.AND.MM.GE.2) GO TO 1777
      READ(5,561)NCQ
  561 FORMAT(I5)
      WRITE(6,660)NCQ
  660 FORMAT(1H0,5X,'NCQ =',I5,/)
      IF(NCQ.EQ.0) GO TO 1660
```

SOURCE STATEMENT

```
      READ(5,560)(NC(I),BCQ(I),I=1,NCQ)
  560 FORMAT(5(I5,F10.4))
      WRITE(6,661)(I,NC(I),BCQ(I),I=1,NCQ)
  661 FORMAT(1H ,/(5(3X,2I4,F12.4)))
      DO 1664 I=1,NCQ
      K=NC(I)
      B(K,MM)=B(K,MM)+BCQ(I)
 1664 CONTINUE
      GO TO 1660
 1777 DO 1779 I=1,NCQ
      K=NC(I)
 1779 B(K,MM)=B(K,1)
 1660 CONTINUE
 1609 CONTINUE
      IF(KEISAN.EQ.2) GO TO 1667
      IF(IQW.EQ.0) GO TO 1665
 1667 WRITE(6,663)
  663 FORMAT(1H0,10X,'B(I,MM) VECTOR',//)
      DO 1666 I=1,NP
      WRITE(6,664)(I,(B(I,MM),MM=1,ME))
  664 FORMAT(1H ,5X,I5,5F12.3)
 1666 CONTINUE
 1665 CONTINUE
      GO TO 100
    7 WRITE(6,601)ID,(MOJI(I),I=1,5)
      IF(KINDV.EQ.0) GO TO 100
      IVEL=ID1
      IVG=ID2
      IVEX=ID3
      IVW=ID4
      IEXP=ID5
      WRITE(6,670)IVEL,IVG,IVEX,IVW,IEXP
  670 FORMAT(1H0,3X,6HIVEL =,I3,5X,5HIVG =,I3,5X,6HIVEX =,I3,5X,
     1       'IVW =',I5,5X,'IEXP =',I5,/)
      GO TO(71,72,73,74,75,76,77,78,79),IVEL
   71 GO TO 100
   72 READ(5,572)((VN(I,J),J=1,IFD),I=1,NP)
  572 FORMAT(4(2F10.0))
      WRITE(6,675)(I,(VN(I,J),J=1,IFD),I=1,NP)
  675 FORMAT(1H ,/(4(3X,I4,1X,2F10.4)))
  721 DO 722 I=1,NE
      DO 722 IZ=1,IFD
      VE(I,IZ)=0.0
      DO 727 J=1,INK
      II=NOD(I,J)
  727 VE(I,IZ)=VE(I,IZ)+VN(II,IZ)
      VE(I,IZ)=VE(I,IZ)/FLOAT(INK)
  722 CONTINUE
      IF(IVW .EQ.0) GO TO 100
      WRITE(6,676)(I,(VE(I,J),J=1,IFD),I=1,NE)
  676 FORMAT(1H ,/(5X,4(I5,2F10.4,3X)))
      GO TO 100
   73 GO TO 100
   74 READ(5,574)(IVNGS(I),IVNGE(I),(VNG(I,J),J=1,IFD),I=1,IVG)
  574 FORMAT(2(2I5,2F10.0))
      WRITE(6,679)(I,IVNGS(I),IVNGE(I),(VNG(I,J),J=1,IFD),I=1,IVG)
```

SOURCE STATEMENT

```
  679 FORMAT(1H ,/(2(3X,I4,1X,2I4,2F10.4)))
      DO 720 I=1,IVG
      IS=IVNGS(I)
      IE=IVNGE(I)
      DO 720 K=IS,IE
      DO 720 J=1,IFD
      VN(K,J)=VNG(I,J)
  720 CONTINUE
      IF(IVW.EQ.0) GO TO 721
      WRITE(6,6701)(I,(VN(I,J),J=1,IFD),I=1,NP)
 6701 FORMAT(1H ,/(5X,4(I4,2F10.4,3X)))
      GO TO 721
   75 GO TO 100
   76 GO TO 100
   77 GO TO 100
   78 GO TO 100
   79 GO TO 100
    8 WRITE(6,601)ID,(MOJI(I),I=1, 5)
      IFL=0
      IF(IFK.EQ.3) GO TO 100
  830 READ(5,587)ID,LD,LD1,LD2
  587 FORMAT(I2,2X,I1,2I5)
      IF(ID.NE.8) GO TO 110
      IF(LD.LE.0.OR.LD.GE.7) GO TO 110
      GO TO (821,822,823,824,825,826),LD
  821 WRITE(6,6821)LD
 6821 FORMAT(1H0,//,5X,'*  8- ',I1,' *',//)
      NBN=LD1
      JBN=LD2
      WRITE(6,680)NBN,JBN
  680 FORMAT(1H0,3X,'NBN =',I5,5X,'JBN =',I5, /)
      IF(NBN.EQ.0.AND.IFK.NE.4) GO TO 87
      IF(NBN.EQ.0.AND.IFK.EQ.4) GO TO 830
      IF(JBN.NE.0) GO TO 80
      READ(5,580)(NF(K),BL(K),K=1,NBN)
  580 FORMAT(5(I5,F10.0))
      WRITE(6,683)(K,NF(K),BL(K),K=1,NBN)
  683 FORMAT(1H ,/(5(3X,I4,I4,F10.4)))
      DO 8000 K=1,NBN
      DO 8000 MM=1,ME
 8000 BV(K,MM)=BL(K)
      GO TO 8001
   80 DO 800 K=1,NBN
      READ(5,583)(NF(K),(BV(K,MM),MM=1,ME))
  583 FORMAT(I5,5X,5F10.0)
      WRITE(6,681)(K,NF(K),(BV(K,MM),MM=1,ME))
  681 FORMAT(1H ,3X,2I5,5F10.4)
  800 CONTINUE
 8001 CONTINUE
      GO TO 830
   87 WRITE(6,6838)
 6838 FORMAT(1H0,5X,25HNBN = 0 DE KEISAN DEKINAI,/)
      GO TO 830
  822 WRITE(6,6821)LD
      IFL=2
      NBCE=LD1
```

SOURCE STATEMENT

```
      WRITE(6,6822)NBCE
 6822 FORMAT(1H0,3X,'NBCE =',I5,/)
   84 IF(NBCE.EQ.0) GO TO 100
      READ(5,584)(NBE(I),I=1,NBCE)
  584 FORMAT(16I5)
      WRITE(6,684)
  684 FORMAT(1H0,/,5X,'** FLUX AND LOSS **',//)
      WRITE(6,685)(I,NBE(I),I=1,NBCE)
  685 FORMAT(1H ,/(3X,16(I3,I4)))
      GO TO 830
  823 WRITE(6,6821)LD
      IFL=3
      NBCL=LD1
      WRITE(6,6823)NBCL
 6823 FORMAT(1H0,3X,'NBCL =',I5,/)
      IF(NBCL.EQ.0) GO TO 100
      READ(5,585)((NBL(I,J),J=1,3),I=1,NBCL)
  585 FORMAT(5(3I5))
      WRITE(6,687)(I,(NBL(I,J),J=1,3),I=1,NBCL)
  687 FORMAT(1H ,/(3X,5(I4,3I4,1X)))
      GO TO 830
  824 WRITE(6,6821)LD
      IFL=4
      DO 8036 I=1,NBCL
      READ(5,5889)(FLUX(I,MM),MM=1,ME)
 5889 FORMAT(5F10.0)
      WRITE(6,6889)I,(FLUX(I,MM),MM=1,ME)
 6889 FORMAT(1H ,3X,I5,5F12.4)
 8036 CONTINUE
      GO TO 830
  825 WRITE(6,6821)LD
      IFL=5
      JCAT=LD1
      WRITE(6,6825)JCAT
 6825 FORMAT(1H0,3X,'JCAT =',I5,/)
      READ(5,586)(CLOSS(I),ATZERO(I),TZERO(I),I=1,NBCL)
  586 FORMAT(2(3F10.0))
      WRITE(6,688)(I,CLOSS(I),ATZERO(I),TZERO(I),I=1,NBCL)
  688 FORMAT(1H ,/(3X,2(I5,3F12.5)))
      GO TO 830
  826 WRITE(6,6821)LD
      IF(IFL.EQ.0) GO TO 100
      IF(NBCE.EQ.0.OR.NBCL.EQ.0) GO TO 100
      DO 8010 I=1,NBCL
      I1=NBL(I,1)
      I2=NBL(I,2)
      I3=NBL(I,3)
      DLS=0.0
      DO 8011 J=1,IFD
 8011 DLS=DLS+(Z(I1,J)-Z(I3,J))**2
      DLS=SQRT(DLS)
      DO 8081 MM=1,ME
      FLAL=(FLUX(I,MM)-ATZERO(I)*TZERO(I))*DLS/6.0
      B(I1,MM)=B(I1,MM)-FLAL
      B(I2,MM)=B(I2,MM)-FLAL*4.0
 8081 B(I3,MM)=B(I3,MM)-FLAL
```

SOURCE STATEMENT

```
 8010 CONTINUE
      GO TO 100
    9 WRITE(6,602)ID,(MOJI(I),I=1, 5)
      KEISAN=ID1
      IBCH=ID2
      IAB=ID3
      IACH=ID4
      IPTE=ID5
      WRITE(6,690)KEISAN,IBCH,IAB,IACH,IPTE
  690 FORMAT(1H0,3X,'KEISAN =',I3,5X,'IBCH =',I3,5X,'IAB =',I5,5X,
     1     'IACH =',I5,5X,'IPTE =',5X,I5,/)
      IF(IBCH.EQ.0) GO TO 99
      WRITE(6,6905)
 6905 FORMAT(1H0,//)
      READ(5,590)(IBCHEK(I),I=1,IBCH)
      WRITE(6,691)(I,IBCHEK(I),I=1,IBCH)
   99 CONTINUE
      IF(IAB.EQ.0) GO TO 956
      DO 90 M=1,NE
   90 NAB(M)=0
      READ(5,590)(NEAB(I),I=1,IAB)
  590 FORMAT(15I5)
      WRITE(6,691)(I,NEAB(I),I=1,IAB)
  691 FORMAT(1H ,/(3X,15(I3,I4,1X)))
      DO 91 I=1,IAB
      NEABE=NEAB(I)
      NAB(NEABE)=1
   91 CONTINUE
  956 CONTINUE
      GO TO 100
   10 WRITE(6,602)ID,MOJI
      JAXB=ID1
      JEXP=ID2
      WRITE( 6,6187)JAXB,JEXP
 6187 FORMAT(1H0,5X,'JAXB =',I3,5X,'JEXP =',I5,/)
      GO TO 100
   11 CONTINUE
      WRITE(6,6119)ID,MOJI
 6119 FORMAT(1H1,//,3X,'* ',I2,' *',5X,5A4,//)
      IF(IFK.NE.4) GO TO 100
      CALL LPREAD(MOJI,NP)
      GO TO 100
   12 WRITE(6,602)ID,MOJI
      IPR=ID1
      JFIG=ID2
      WRITE(6,6612)IPR,JFIG
 6612 FORMAT(1H0,3X,'IPR =',I3,5X,'JFIG =',I3,/)
      IF(IPR.EQ.0) GO TO 100
      CALL PF26
      GO TO 100
   13 WRITE(6,6199)ID
 6199 FORMAT(1H0,//,2X,2H* ,I2,2H *,5X,11HDATA FINISH,/)
      GO TO 1001
  110 WRITE(6,6110)ID
 6110 FORMAT(1H0,//,2X,2H* ,I2,2H *,5X,9HID OKASII,/)
      STOP
```

```
SOURCE STATEMENT

 1001 RETURN
      END
```

SOURCE STATEMENT

```
C     SUBROUTINE
      SUBROUTINE AUT
      DIMENSION ZX(30),ZY(30),NOD4(9)
      DIMENSION Z1(30,2),Z2(30,2),Z3(30,2),Z4(30,2)
      DIMENSION MXTEI(30),MYTEI(30),MXT(50),MYT(50)
      COMMON/SABCDX/DELTA( 500)
      COMMON/DATA0R/NF331,NF332,NF333,IEK,INK,IFK,IFD,KNI
      COMMON/DATA2R/Z(500,2),NP
      COMMON/DATA3R/AREAS,SEIDO,NE,INKE,IAREA,ISEIDO,NOD(500,6)
      COMMON/SAUT/  XSTART,YSTART,XEND,YEND,DMX,DMY,MXE,MYE
      READ(5,500)ISHAPE,INEXY,NANAME,ITEI,IWR,ITEIZ,ITEIEN
  500 FORMAT(5X,7I5)
      WRITE(6,600)ISHAPE,INEXY,NANAME,ITEI,IWR,ITEIZ,ITEIEN
  600 FORMAT(1H0,3X,8HISHAPE =,I3,5X,7HINEXY =,I3,5X,8HNANAME =,I3,5X,
     1  6HITEI =,I3,5X,5HIWR =,I3,5X,7HITEIZ =,I3,5X,8HITEIEN =,I3,/)
      IF(ITEI.EQ.0) GO TO 100
      READ(5,501)(MXTEI(I),MYTEI(I),I=1,ITEI)
  501 FORMAT(16I5)
      WRITE(6,601)(I,MXTEI(I),MYTEI(I),I=1,ITEI)
  601 FORMAT(1H0,3X,13HNANAME TEISEI,/(8(3X,I2,2I4)))
  100 GO TO (11,12,13,14,15,16,17,18),ISHAPE
   11 WRITE(6,611)
  611 FORMAT(1H0,/,5X,16HTOOKANKAKU KOUSI,/)
      READ(5,502)KXE,KYE,XSTART,YSTART,XEND,YEND
  502 FORMAT(2I5,4F10.0)
      WRITE(6,6110)KXE,KYE,XSTART,YSTART,XEND,YEND
 6110 FORMAT(1H0,3X,5HKXE =,I4,5X,5HKYE =,I4,5X,8HXSTART =,F10.4,5X,
     1       8HYSTART =,F10.4,5X,6HXEND =,F10.4,5X,6HYEND =,F10.4,/)
      MXE=KXE-1
      MYE=KYE-1
      DMX=(XEND-XSTART)/FLOAT(MXE)
      DMY=(YEND-YSTART)/FLOAT(MYE)
      NNPP=KXE*KYE
      NP=(2*KYE-1)*(2*KXE-1)
      NE4=MXE*MYE
      NE=2*NE4
      ZY(1)=YSTART
      ZX(1)=XSTART
      DO 3 I=2,KYE
      I1=I-1
    3 ZY(I)=ZY(I1)+DMY
      DO 2 J=2,KXE
      J1=J-1
    2 ZX(J)=ZX(J1)+DMX
  200 DO 4 I=1,KYE
      DO 4 J=1,KXE
      IF(INEXY.NE.0) GO TO 20
      K=(I-1)*(2*KXE-1)*2+2*J-1
      GO TO 202
   20 K=(J-1)*(2*KYE-1)*2+2*I-1
  202 Z(K,1)=ZX(J)
    4 Z(K,2)=ZY(I)
  203 DO 390 I=1,MYE
  390 MYT(I)=0
      DO 391 J=1,MXE
  391 MXT(J)=0
```

SOURCE STATEMENT

```
      IF(ITEI.EQ.0) GO TO 393
      DO 392 I=1,ITEI
      MTX=MXTEI(I)
      MTY=MYTEI(I)
      MXT(MTX)=1
  392 MYT(MTY)=1
  393 CONTINUE
      DO 6 I=1,MYE
      IYTEI=MYT(I)
      DO 6 J=1,MXE
      IXTEI=MXT(J)
      IF(ISHAPE.EQ.5) GO TO 244
      IF(INEXY.NE.0) GO TO 244
      M=(I-1)*MXE+J
      I1=M*2-1
      I2=M*2
      NOD4(1)=(I-1)*(2*KXE-1)*2+2*J-1
      NOD4(2)=NOD4(1)+2
      NOD4(3)=NOD4(1)+(2*KXE-1)*2+2
      NOD4(4)=NOD4(3)-2
      NOD4(5)=NOD4(1)+1
      NOD4(6)=NOD4(1)+(2*KXE-1)
      NOD4(7)=NOD4(6)+1
      NOD4(8)=NOD4(7)+1
      NOD4(9)=NOD4(4)+1
      GO TO 245
  244 M=(J-1)*MYE+I
      I1=M*2-1
      I2=M*2
      NOD4(1)=(J-1)*(2*KYE-1)*2+2*I-1
      NOD4(2)=NOD4(1)+(2*KYE-1)*2
      NOD4(3)=NOD4(2)+2
      NOD4(4)=NOD4(1)+2
      NOD4(5)=NOD4(1)+(2*KYE-1)
      NOD4(6)=NOD4(1)+1
      NOD4(7)=NOD4(5)+1
      NOD4(8)=NOD4(2)+1
      NOD4(9)=NOD4(7)+1
  245 CONTINUE
      IF(IWR.LT.1) GO TO 1066
      WRITE(6,6820) I,J,M,(NOD4(IK),IK=1,9)
 6820 FORMAT(1H ,5X,3I5,5X,9I5)
 1066 CONTINUE
      MODI2=MOD(I,2)
      MODJ2=MOD(J,2)
      GO TO (21,22,23,24),NANAME
   21 IF(MODI2.NE.0.AND.MODJ2.NE.0) GO TO 8
      IF(MODI2.EQ.0.AND.MODJ2.EQ.0) GO TO 8
      GO TO 9
    8 IF(IYTEI.NE.0.AND.IXTEI.NE.0) GO TO 90
   80 NOD(I1,1)=NOD4(   1)
      NOD(I1,2)=NOD4(   3)
      NOD(I1,3)=NOD4(   4)
      NOD(I1,4)=NOD4(9)
      NOD(I1,5)=NOD4(6)
      NOD(I1,6)=NOD4(7)
```

SOURCE STATEMENT

```
      NOD(I2,1)=NOD4(  1)
      NOD(I2,2)=NOD4(  2)
      NOD(I2,3)=NOD4(  3)
      NOD(I2,4)=NOD4(8)
      NOD(I2,5)=NOD4(7)
      NOD(I2,6)=NOD4(5)
      GO TO 6
    9 IF(IYTEI.NE.0.AND.IXTEI.NE.0) GO TO 80
   90 NOD(I1,1)=NOD4(  1)
      NOD(I1,2)=NOD4(  2)
      NOD(I1,3)=NOD4(  4)
      NOD(I1,4)=NOD4(7)
      NOD(I1,5)=NOD4(6)
      NOD(I1,6)=NOD4(5)
      NOD(I2,1)=NOD4(  2)
      NOD(I2,2)=NOD4(  3)
      NOD(I2,3)=NOD4(  4)
      NOD(I2,4)=NOD4(9)
      NOD(I2,5)=NOD4(7)
      NOD(I2,6)=NOD4(8)
      GO TO 6
   22 IF(MODI2.NE.0.AND.MODJ2.EQ.0) GO TO 8
      IF(MODI2.EQ.0.AND.MODJ2.NE.0) GO TO 8
      GO TO 9
   23 GO TO 8
   24 GO TO 9
    6 CONTINUE
      WRITE(6,602) NP,NNPP,NE,NE4
  602 FORMAT(1H0,5X,4HNP =,I5,5X,'NNPP =',I5,5X,4HNE =,I5,5X,5HNE4 =,I5,
     1      /)
      IF(IWR.EQ.0) GO TO 300
      WRITE(6,605)(I,(NOD(I,J),J=1,6),I=1,NE)
  605 FORMAT(1H1,5X,18HSANKAKUKEI ELEMENT,//(5X,2(I4,6I5,1X)))
   31 CONTINUE
  300 IF(INEXY.NE.0) GO TO 780
      DO 782 I=1,KYE
      DO 782 J=1,KXE
      K=(I-1)*(2*KXE-1)*2+2*J-1
  782 WRITE(6,6781)I,J,K,(Z(K,JJ),JJ=1,2)
      GO TO 781
  780 DO 783 J=1,KXE
      DO 783 I=1,KYE
      K=(J-1)*(2*KYE-1)*2+2*I-1
      WRITE(6,6781)J,I,K,(Z(K,JJ),JJ=1,2)
 6781 FORMAT(1H ,10X,3I5,2F10.2)
  783 CONTINUE
  781 CONTINUE
      IF(IAREA.NE.-2) GO TO 30
      VOL=0.0
      DO 85 M=1,NE
      CALL ABCDXY(M)
      VOL=VOL+DELTA(M)
      WRITE(6,619)M,VOL,DELTA(M)
  619 FORMAT(1H ,20X,I5,2F12.4)
   85 CONTINUE
      GO TO 30
```

SOURCE STATEMENT

```
   12 GO TO 30
   13 GO TO 30
   14 WRITE(6,6140)
 6140 FORMAT(1H0,/,5X,18HFUTOOKANKAKU KOUSI,/)
      READ(5,5140)KXE,KYE
 5140 FORMAT(2I5)
      WRITE(6,6141)KXE,KYE
 6141 FORMAT(1H0,3X,5HKXE =,I4,5X,5HKYE =,I4,/)
      READ(5,5141)(ZX(J),J=1,KXE)
 5141 FORMAT(8F10.0)
      WRITE(6,6142)(J,ZX(J),J=1,KXE)
 6142 FORMAT(1H0,/(8(2X,I2,F10.4)))
      READ(5,5142)(ZY(I),I=1,KYE)
 5142 FORMAT(8F10.0)
      WRITE(6,6143)(I,ZY(I),I=1,KYE)
 6143 FORMAT(1H0,/(8(2X,I2,F10.4)))
      MXE=KXE-1
      MYE=KYE-1
      NP=(2*KYE-1)*(2*KXE-1)
      NNPP=KXE*KYE
      NE4=MXE*MYE
      NE=2*NE4
      GO TO 200
   15 WRITE(6,6150))
 6150 FORMAT(1H0,/,5X,19HFUTOOHEN KOUSI (A) ,/)
      READ(5,5150)KXE,KYE
 5150 FORMAT(2I5)
      WRITE(6,6151)KXE,KYE
 6151 FORMAT(1H0,3X,5HKXE =,I4,5X,5HKYE =,I4,/)
      READ(5,5151)((Z1(I,J),J=1,2),I=1,KXE)
 5151 FORMAT(4(2F10.0))
      WRITE(6,6152)(I,(Z1(I,J),J=1,2),I=1,KXE)
 6152 FORMAT(1H0,/(4(3X,I4,1X,2F10.4)))
      READ(5,5152)((Z2(I,J),J=1,2),I=1,KYE)
 5152 FORMAT(4(2F10.0))
      WRITE(6,6153)(I,(Z2(I,J),J=1,2),I=1,KYE)
 6153 FORMAT(1H0,/(4(3X,I4,1X,2F10.4)))
      READ(5,5153)((Z3(I,J),J=1,2),I=1,KXE)
 5153 FORMAT(4(2F10.0))
      WRITE(6,6154)(I,(Z3(I,J),J=1,2),I=1,KXE)
 6154 FORMAT(1H0,/(4(3X,I4,1X,2F10.4)))
      READ(5,5154)((Z4(I,J),J=1,2),I=1,KYE)
 5154 FORMAT(4(2F10.0))
      WRITE(6,6155)(I,(Z4(I,J),J=1,2),I=1,KYE)
 6155 FORMAT(1H0,/(4(3X,I4,1X,2F10.4)))
      MXE=KXE-1
      MYE=KYE-1
      NNPP=KXE*KYE
      NP=(2*KYE-1)*(2*KXE-1)
      NE4=MXE*MYE
      NE=2*NE4
      DO 150 I=1,KYE
      K=(1-1)*(2*KYE-1)*2+I*2-1
      Z(K,1)=Z4(I,1)
  150 Z(K,2)=Z4(I,2)
      DO 151 J=2,MXE
```

SOURCE STATEMENT

```
      DY=(Z3(J,2)-Z1(J,2))/FLOAT(MYE)
      DX=(Z3(J,1)-Z1(J,1))/FLOAT(MYE)
      DO 152 I=1,KYE
      K=(J-1)*(2*KYE-1)*2+2*I-1
      Z(K,1)=Z1(J,1)+(I-1)*DX
  152 Z(K,2)=Z1(J,2)+(I-1)*DY
  151 CONTINUE
      DO 153 I=1,KYE
      K=(KXE-1)*(2*KYE-1)*2+I*2-1
      Z(K,1)=Z2(I,1)
  153 Z(K,2)=Z2(I,2)
      GO TO 203
   16 GO TO 30
   17 GO TO 30
   18 GO TO 30
   30 RETURN
      END
```

SOURCE STATEMENT

```
C     SUBROUTINE
       SUBROUTINE ABCDXY(M)
       COMMON/SABCDX/DELTA( 500)
       COMMON/DATA2R/Z(500,2),NP
       COMMON/DATA3R/AREAS,SEIDO,NE,INKE,IAREA,ISEIDO,NOD(500,6)
       II=NOD(M,1)
       IJ=NOD(M,2)
       IM=NOD(M,3)
       XI=Z(II,1)
       YI=Z(II,2)
       XJ=Z(IJ,1)
       YJ=Z(IJ,2)
       XM=Z(IM,1)
       YM=Z(IM,2)
       GX=(Z(II,1)+Z(IJ,1)+Z(IM,1))/3.0
       GY=(Z(II,2)+Z(IJ,2)+Z(IM,2))/3.0
       XI=XI-GX
       XJ=XJ-GX
       XM=XM-GX
       YI=YI-GY
       YJ=YJ-GY
       YM=YM-GY
       DEL=XJ*YM+XI*YJ+XM*YI-XJ*YI-XI*YM-XM*YJ
       DELTA(M)=DEL*0.5
       RETURN
       END
```

SOURCE STATEMENT

```
C     SUBROUTINF
       SUBROUTINE SKMT(M)
       DIMENSION SH(6,6)
       DIMENSION L(3),L1(3),L2(3),L3(3),L4(3),L5(3),L6(3),L7(3)
       DIMENSION DXK(2)
       DIMENSION SN(6,6),XKSN(6,6,2),XSN(6,6)
       DIMENSION FNXNG(6,6,2)
       DIMENSION BK(2,3)
       DIMENSION N2(3,2)
       COMMON/FKIND/ KIND,KINDD,KINDK,KINDV
       COMMON/DATA0R/NF331,NF332,NF333,IEK,INK,IFK,IFD,KNI
       COMMON/DATA1R/IAD,KFEM1,MTITL( 5)
       COMMON/DATA2R/Z(500,2),NP
       COMMON/DATA3R/AREAS,SEIDO,NE,INKE,IAREA,ISEIDO,NOD(500,6)
       COMMON/DATA4 /BU(50,8)
       COMMON/DATA5C/KBUK(500)
       COMMON/DATA7R/VE(500,2),VN(500,2),IVEL,IVG,IVEX,IVW,IEXP
       COMMON/DATA8B/FLUX(300,5),CLOSS(300),ATZERO(300),TZERO(300)
      1      ,NBE(300),NBL(300,3),NBCE,NBCL,JCAT
       COMMON/SSKMT/HH(6,6)
       COMMON/DATA9R/KEISAN,IAB,NAB(500)
       COMMON/SABCDX/DELTA( 500)
       COMMON/DATAHF/IHF
       COMMON/FCHECK/I,J
       INK1=INK+1
       NEAM=KBUK(M)
       DXK(1)=BU(NEAM,1)
       DXK(2)=BU(NEAM,2)
       DK=BU(NEAM,3)
       DT=2.0*DELTA(M)
       DEL2=DT*DT
       II=NOD(M,1)
       IJ=NOD(M,2)
       IM=NOD(M,3)
       XI=Z(II,1)
       YI=Z(II,2)
       XJ=Z(IJ,1)
       YJ=Z(IJ,2)
       XM=Z(IM,1)
       YM=Z(IM,2)
       GX=(Z(II,1)+Z(IJ,1)+Z(IM,1))/3.0
       GY=(Z(II,2)+Z(IJ,2)+Z(IM,2))/3.0
       XI=XI-GX
       XJ=XJ-GX
       XM=XM-GX
       YI=YI-GY
       YJ=YJ-GY
       YM=YM-GY
       BK(1,1)=YJ-YM
       BK(1,2)=YM-YI
       BK(1,3)=YI-YJ
       BK(2,1)=XM-XJ
       BK(2,2)=XI-XM
       BK(2,3)=XJ-XI
C      I=1,INK     J=1,INK
       DO 2 I=1,INK
```

SOURCE STATEMENT

```
      IF(KINDV.EQ.0.AND.KINDK.EQ.0) GO TO 5
      L1(I)=0
      L2(I)=0
      L3(I)=0
    5 L4(I)=0
      L5(I)=0
      L6(I)=0
      L7(I)=0
    2 CONTINUE
      CALL FABCDS(M,L7,FS)
      FS7=FS
      DO 1 I=1,INK
      IF(KINDV.EQ.0.AND.KINDK.EQ.0) GO TO 6
      L1(I)=2
      L2(I)=2
      L3(I)=1
    6 L4(I)=1
      L5(I)=1
      CALL FABCDS(M,L5,FS)
      FS5=FS
      DO 3 J=1,INK
      IF(KINDV.EQ.0.AND.KINDK.EQ.0) GO TO 7
      L1(J)=L1(J)+2
      CALL FABCDS(M,L1,FS)
      FS1=FS
      L2(J)=L2(J)+1
      CALL FABCDS(M,L2,FS)
      FS2=FS
      L3(J)=L3(J)+2
      CALL FABCDS(M,L3,FS)
      FS3=FS
    7 L4(J)=L4(J)+1
      CALL FABCDS(M,L4,FS)
      FS4=FS
      L6(J)=L6(J)+1
      CALL FABCDS(M,L6,FS)
      FS6=FS
      IF(KINDV.EQ.0.AND.KINDK.EQ.0) GO TO 8
      SN(I,J)=4.0*FS1-2.0*FS2-2.0*FS3+FS4
    8 XSN(I,J)=16.0*FS4-4.0*FS5-4.0*FS6+FS7
      DO 9 K=1,IFD
      XKSN(I,J,K)=XSN(I,J)*BK(K,I)*BK(K,J)/DEL2
    9 CONTINUE
      IF(KINDV.EQ.0.AND.KINDK.EQ.0) GO TO 10
      L1(J)=L1(J)-2
      L2(J)=L2(J)-1
      L3(J)=L3(J)-2
   10 L4(J)=L4(J)-1
      L6(J)=L6(J)-1
    3 CONTINUE
      IF(KINDV.EQ.0.AND.KINDK.EQ.0) GO TO 11
      L1(I)=L1(I)-2
      L2(I)=L2(I)-2
      L3(I)=L3(I)-1
   11 L4(I)=L4(I)-1
      L5(I)=L5(I)-1
```

SOURCE STATEMENT

```
    1 CONTINUE
C     I=1,INK      J=INK1,KNI
      N2(1,1)=2
      N2(1,2)=3
      N2(2,1)=3
      N2(2,2)=1
      N2(3,1)=1
      N2(3,2)=2
      DO 20 I=1,INK
      IF(KINDV.EQ.0.AND.KINDK.EQ.0) GO TO 28
      L1(I)=0
      L2(I)=0
   28 L3(I)=0
      L4(I)=0
   20 CONTINUE
      DO 21 I=1,INK
      IF(KINDV.EQ.0.AND.KINDK.EQ.0)  GO  TO  26
      L1(I)=2
      L2(I)=1
   26 L3(I)=1
      L4(I)=1
      DO 22 J=INK1,KNI
      J4=J-INK
      NJ1=N2(J4,1)
      NJ2=N2(J4,2)
      IF(KINDV.EQ.0.AND.KINDK.EQ.0) GO TO 23
      L1(NJ1)=L1(NJ1)+1
      L1(NJ2)=L1(NJ2)+1
      L2(NJ1)=L2(NJ1)+1
      L2(NJ2)=L2(NJ2)+1
      CALL FABCDS(M,L1,FS)
      FS1=FS
      CALL FABCDS(M,L2,FS)
      FS2=FS
      SN(I,J)=8.0*FS1-4.0*FS2
      L1(NJ1)=L1(NJ1)-1
      L1(NJ2)=L1(NJ2)-1
      L2(NJ1)=L2(NJ1)-1
      L2(NJ2)=L2(NJ2)-1
   23 L3(NJ2)=L3(NJ2)+1
      L4(NJ1)=L4(NJ1)+1
      L5(NJ2)=1
      L6(NJ1)=1
      CALL FABCDS(M,L3,FS)
      FS3=FS
      CALL FABCDS(M,L4,FS)
      FS4=FS
      CALL FABCDS(M,L5,FS)
      FS5=FS
      CALL FABCDS(M,L6,FS)
      FS6=FS
      DO 25 K=1,IFD
      XKSN(I,J,K)=4.0/DEL2*(4.0*BK(K,I)*BK(K,NJ1)*FS3+4.0*BK(K,I)*
     1               BK(K,NJ2)*FS4-BK(K,I)*BK(K,NJ1)*FS5
     2              -BK(K,I)*BK(K,NJ2)*FS6)
   25 CONTINUE
```

SOURCE STATEMENT

```
      L3(NJ2)=L3(NJ2)-1
      L4(NJ1)=L4(NJ1)-1
      L5(NJ2)=L5(NJ2)-1
      L6(NJ1)=L6(NJ1)-1
   22 CONTINUE
      IF(KINDV.EQ.0.AND.KINDK.EQ.0) GO TO 24
      L1(I)=L1(I)-2
      L2(I)=L2(I)-1
   24 CONTINUE
      L3(I)=L3(I)-1
      L4(I)=L4(I)-1
   21 CONTINUE
C     I=INK1,KNI       J=1,INK
      DO 30 I=INK1,KNI
      DO 30 J=1,INK
      IF(KINDV.EQ.0.AND.KINDK.EQ.0) GO TO 33
      SN(I,J)=SN(J,I)
   33 DO 31 K=1,IFD
   31 XKSN(I,J,K)=XKSN(J,I,K)
   30 CONTINUE
C     I=INK1,KNI       J=INK1,KNI
      DO 40 I=1,INK
   40 L(I)=0
      DO 41 I=INK1,KNI
      I4=I-INK
      NI1=N2(I4,1)
      NI2=N2(I4,2)
      IF(KINDV.EQ.0.AND.KINDK.EQ.0) GO TO 44
      L(NI1)=1
      L(NI2)=1
   44 CONTINUE
      L1(NI2)=1
      L2(NI2)=1
      L3(NI1)=1
      L4(NI1)=1
      DO 42 J=INK1,KNI
      J4=J-INK
      NJ1=N2(J4,1)
      NJ2=N2(J4,2)
      IF(KINDV.EQ.0.AND.KINDK.EQ.0) GO TO 43
      L(NJ1)=L(NJ1)+1
      L(NJ2)=L(NJ2)+1
      CALL FABCDS(M,L,FS)
      SN(I,J)=16.0*FS
      L(NJ1)=L(NJ1)-1
      L(NJ2)=L(NJ2)-1
   43 L1(NJ2)=L1(NJ2)+1
      L2(NJ1)=L2(NJ1)+1
      L3(NJ2)=L3(NJ2)+1
      L4(NJ1)=L4(NJ1)+1
      CALL FABCDS(M,L1,FS)
      FS1=FS
      CALL FABCDS(M,L2,FS)
      FS2=FS
      CALL FABCDS(M,L3,FS)
      FS3=FS
```

SOURCE STATEMENT

```
      CALL FABCDS(M,L4,FS)
      FS4=FS
      L1(NJ2)=L1(NJ2)-1
      L2(NJ1)=L2(NJ1)-1
      L3(NJ2)=L3(NJ2)-1
      L4(NJ1)=L4(NJ1)-1
      DO 45 K=1,IFD
      XKSN(I,J,K)=16.0/DEL2*(BK(K,NI1)*BK(K,NJ1)*FS1+BK(K,NI1)*
     1    BK(K,NJ2)*FS2+BK(K,NI2)*BK(K,NJ1)*FS3+BK(K,NI2)*BK(K,NJ2)*FS4)
   45 CONTINUE
   42 CONTINUE
      IF(KINDV.EQ.0.AND.KINDK.EQ.0) GO TO 47
      L(NI1)=L(NI1)-1
      L(NI2)=L(NI2)-1
   47 CONTINUE
      L1(NI2)=L1(NI2)-1
      L2(NI2)=L2(NI2)-1
      L3(NI1)=L3(NI1)-1
      L4(NI1)=L4(NI1)-1
   41 CONTINUE
      IF(IHF.GE.0) GO TO 80
      CALL FNXN2G(M,BK,N2,DT,FNXNG)
   80 CONTINUE
      DO 150 I=1,KNI
      DO 150 J=1,KNI
      H=0.0
      HN=0.0
      HF=0.0
      HR=0.0
      H=H+DXK(1)*XKSN(I,J,1)+DXK(2)*XKSN(I,J,2)
      IF(KINDV.EQ.0) GO TO 81
      IF(IHF) 199,200,201
  199 HF=HF+(VE(M,1)*FNXNG(I,J,1)+VE(M,2)*FNXNG(I,J,2))
      GO TO 81
  200 HN=HN+(VE(M,1)*VE(M,1)/4.0/DXK(1)+VE(M,2)*VE(M,2)/4.0/DXK(2))*
     1   SN(I,J)
      GO TO 81
  201 CONTINUE
   81 IF(KINDK.EQ.0) GO TO 83
      HR=HR+DK*SN(I,J)
   83 CONTINUE
      HH(I,J)=H+HN+HF+HR
      IF(IAB.EQ.0) GO TO 150
      IF(NAB(M).EQ.0) GO TO 150
      WRITE(6,600)M,I,J,HH(I,J),H,HN,HF,HR
  600 FORMAT(1H ,5X,'M,I,J,HH(I,J),H,HN,HF,HR',5X,3I5,5E12.4)
  150 CONTINUE
      IF(NBCE.EQ.0.AND.NBCL.EQ.0) GO TO 110
      DO 111 I=1,NBCE
      IF(M.EQ.NBE(I)) GO TO 112
  111 CONTINUE
      GO TO 110
  112 CONTINUE
      DO 113 I=1,KNI
      DO 113 J=1,KNI
  113 SH(I,J)=0.0
```

SOURCE STATEMENT

```
      DO 114 K=1,NBCL
      IF(JCAT,EQ.0) CLAT=CLOSS(K)
      IF(JCAT,NE.0) CLAT=ATZERO(K)
      IF(CLAT.EQ.0) GO TO 114
      I1=NBL(K,1)
      I3=NBL(K,3)
      IF(I1.EQ.II.AND.I3.EQ.IJ) GO TU 115
      IF(I1.EQ.II.AND.I3.EQ.IM) GO TU 116
      IF(I1.EQ.IJ.AND.I3.EQ.II) GO TU 115
      IF(I1.EQ.IJ.AND.I3.EQ.IM) GO TO 117
      IF(I1.EQ.IM.AND.I3.EQ.II) GO TU 116
      IF(I1.EQ.IM.AND.I3.EQ.IJ) GO TU 117
      GO TO 114
  115 DLS=SQRT(BK(2,3)**2+BK(1,3)**2)
      AL30 =CLAT*DLS/30.0
      SH(1,1)=AL30*4.0
      SH(1,2)=AL30*(-1.0)
      SH(1,6)=AL30*2.0
      SH(2,1)=AL30*(-1.0)
      SH(2,2)=AL30*4.0
      SH(2,6)=AL30*2.0
      SH(6,1)=AL30*2.0
      SH(6,2)=AL30*2.0
      SH(6,6)=AL30*16.0
      GO TO 114
  116 DLS=SQRT(BK(2,2)**2+BK(1,2)**2)
      AL30 =CLAT*DLS/30.0
      SH(1,1)=AL30*4.0
      SH(1,3)=AL30*(-1.0)
      SH(1,5)=AL30*2.0
      SH(3,1)=AL30*(-1.0)
      SH(3,3)=AL30*4.0
      SH(3,5)=AL30*2.0
      SH(5,1)=AL30*2.0
      SH(5,3)=AL30*2.0
      SH(5,5)=AL30*16.0
      GO TO 114
  117 DLS=SQRT(BK(2,1)**2+BK(1,1)**2)
      AL30=CLAT*DLS/30.0
      SH(2,2)=AL30*4.0
      SH(2,3)=AL30*(-1.0)
      SH(2,4)=AL30*2.0
      SH(3,2)=AL30*(-1.0)
      SH(3,3)=AL30*4.0
      SH(3,4)=AL30*2.0
      SH(4,2)=AL30*2.0
      SH(4,3)=AL30*2.0
      SH(4,4)=AL30*16.0
      DO 118 I=1,KNI
      DO 119 J=1,KNI
  119 HH(I,J)=HH(I,J)+SH(I,J)
      IF(IAB.EQ.0) GO TO 118
      IF(NAB(M).EQ.0) GO TO 118
      WRITE(6,610)M,K,(NBL(K,LL),LL=1,3),I,(SH(I,J),J=1,KNI)
  610 FORMAT(1H ,'M,K,(NBL(K,LL),LL=1,3),I,(SH(I,J),J=1,KNI)',
     1      6I4,6E10.3)
```

SOURCE STATEMENT

```
  118 CONTINUE
  114 CONTINUE
  110 CONTINUE
      RETURN
      END
```

SOURCE STATEMENT

```
C       SUBROUTINE
        SUBROUTINE FNXN2G(M,BK,N2,DT,FNXNG)
        DIMENSION BK(2,3),N2(3,2),FNXNG(6,6,2)
        DIMENSION L1(3),L2(3),L3(3),L4(3)
        COMMON/DATAOR/NF331,NF332,NF333,IEK,INK,IFK,IFD,KNI
        INK1=INK+1
        DO 2 I=1,INK
        L1(I)=0
        L2(I)=0
        L3(I)=0
        L4(I)=0
      2 CONTINUE
C       I=1,INK        J=1,INK
        DO 1 I=1,INK
        L1(I)=2
        L2(I)=1
        L3(I)=2
        L4(I)=1
        CALL FABCDS(M,L3,FS)
        FS3=FS
        CALL FABCDS(M,L4,FS)
        FS4=FS
        DO 3 J=1,INK
        L1(J)=L1(J)+1
        L2(J)=L2(J)+1
        CALL FABCDS(M,L1,FS)
        FS1=FS
        CALL FABCDS(M,L2,FS)
        FS2=FS
        L1(J)=L1(J)-1
        L2(J)=L2(J)-1
        DO 9 K=1,IFD
      9 FNXNG(I,J,K)=(8.0*FS1-4.0*FS2-2.0*FS3+FS4)*BK(K,J)/DT
      3 CONTINUE
        L1(I)=L1(I)-2
        L2(I)=L2(I)-1
        L3(I)=L3(I)-2
        L4(I)=L4(I)-1
      1 CONTINUE
C       I=1,INK        J=INK1,KNI
        DO 20 I=1,INK
        L1(I)=0
        L2(I)=0
        L3(I)=0
     20 L4(I)=0
        DO 21 I=1,INK
        L1(I)=2
        L2(I)=1
        L3(I)=2
        L4(I)=1
        DO 22 J=INK1,KNI
        J4=J-INK
        NJ1=N2(J4,1)
        NJ2=N2(J4,2)
        L1(NJ2)=L1(NJ2)+1
        L2(NJ2)=L2(NJ2)+1
```

SOURCE STATEMENT

```
      L3(NJ1)=L3(NJ1)+1
      L4(NJ1)=L4(NJ1)+1
      CALL FABCDS(M,L1,FS)
      FS1=FS
      CALL FABCDS(M,L2,FS)
      FS2=FS
      CALL FABCDS(M,L3,FS)
      FS3=FS
      CALL FABCDS(M,L4,FS)
      FS4=FS
      DO 23 K=1,IFD
      FNXNG(I,J,K)=((8.0*FS1-4.0*FS2)*BK(K,NJ1)
     1             +(8.0*FS3-4.0*FS4)*BK(K,NJ2))/DT
   23 CONTINUE
      L1(NJ2)=L1(NJ2)-1
      L2(NJ2)=L2(NJ2)-1
      L3(NJ1)=L3(NJ1)-1
      L4(NJ1)=L4(NJ1)-1
   22 CONTINUE
      L1(I)=L1(I)-2
      L2(I)=L2(I)-1
      L3(I)=L3(I)-2
      L4(I)=L4(I)-1
   21 CONTINUE
C     I=INK1,KNI     J=1,INK
      DO 33 I=1,INK
      L1(I)=0
      L2(I)=0
      L3(I)=0
   33 L4(I)=0
      DO 30 I=INK1,KNI
      I4=I-INK
      NI1=N2(I4,1)
      NI2=N2(I4,2)
      L1(NI1)=1
      L1(NI2)=1
      L2(NI1)=1
      L2(NI2)=1
      CALL FABCDS(M,L2,FS)
      FS2=FS
      DO 31 J=1,INK
      L1(J)=L1(J)+1
      CALL FABCDS(M,L1,FS)
      FS1=FS
      DO 32 K=1,IFD
   32 FNXNG(I,J,K)=(16.0*FS1-4.0*FS2)*BK(K,J)/DT
      L1(J)=L1(J)-1
   31 CONTINUE
      L1(NI1)=L1(NI1)-1
      L1(NI2)=L1(NI2)-1
      L2(NI1)=L2(NI1)-1
      L2(NI2)=L2(NI2)-1
   30 CONTINUE
C     I=INK1,KNI     J=INK1,KNI
      DO 43 I=1,INK
      L1(I)=0
```

SOURCE STATEMENT

```
      L2(I)=0
      L3(I)=0
   43 L4(I)=0
      DO 40 I=INK1,KNI
      I4=I-INK
      NI1=N2(I4,1)
      NI2=N2(I4,2)
      L1(NI1)=1
      L1(NI2)=1
      L2(NI1)=1
      L2(NI2)=1
      DO 41 J=INK1,KNI
      J4=J-INK
      NJ1=N2(J4,1)
      NJ2=N2(J4,2)
      L1(NJ2)=L1(NJ2)+1
      L2(NJ1)=L2(NJ1)+1
      CALL FABCDS(M,L1,FS)
      FS1=FS
      CALL FABCDS(M,L2,FS)
      FS2=FS
      DO 42 K=1,IFD
   42 FNXNG(I,J,K)=16.0*(FS1*BK(K,NJ1)+FS2*BK(K,NJ2))/DT
      L1(NJ2)=L1(NJ2)-1
      L2(NJ1)=L2(NJ1)-1
   41 CONTINUE
      L1(NI1)=L1(NI1)-1
      L1(NI2)=L1(NI2)-1
      L2(NI1)=L2(NI1)-1
      L2(NI2)=L2(NI2)-1
   40 CONTINUE
      RETURN
      END
```

SOURCE STATEMENT

```
C     SUBROUTINE
      SUBROUTINE FABCDS(M,L,FS)
      COMMON/FCHECK/IF1,IF2
      COMMON/SABCDX/DELTA( 500)
      DIMENSION L(3),FLU(3)
      TWO=2
      FLABCD=1
      LL=0
      DO 1 I=1,3
      LL=LL+L(I)
      FLU(I)=1
      LI=L(I)
      IF(LI.LE.1) GO TO 1
      DO 2 K=1,LI
      FK=FLOAT(K)
    2 FLU(I)=FLU(I)*FK
      FLABCD=FLABCD*FLU(I)
    1 CONTINUE
      L3=LL+2
      FABCD3=1
      DO 3 K=1,L3
      FK=FLOAT(K)
    3 FABCD3=FABCD3*FK
      FS=FLABCD/FABCD3*TWO*DELTA(M)
    5 CONTINUE
      RETURN
      END
```

```
SOURCE STATEMENT

C      SUBROUTINE
       SUBROUTINE NKE
       RETURN
       END
```

SOURCE STATEMENT

```
C     SUBROUTINF
       SUBROUTINE PF26
       RETURN
       END
```

SOURCE STATEMENT

```
C     SUBROUTINE
      SUBROUTINE LIPR
C 199 HI/TC/LP02 LINEAR PROGRAMMING BY SIMPLEX METHOD
      COMMON/SLP1  /A(345,355),B(500),CC(500)
      COMMON/SLP2  /M1,M2,M3,N0,NAME(5),JLPW,NITERE
      COMMON/SLP3  /BP(500),C(500),IB(500),NUM(500)
      COMMON/SLP4  /Z,ZERO,ONE,PGRE,EPS
      COMMON/SLP5  /M,N,NC,N1F(11),N2F(5),LS,NAX,MAX
      COMMON/SLP6  /CB(450),CCC(450)
      COMMON/SLP7  /XL(200),NXLS(200),NXLE(200),NFUS(200),NFUE(200)
      COMMON/SLP8  /LBXL(200),LBXLE,KFU(200),KFUE
      COMMON/SLP9  /MXLG,NFUG,NFUN,LPDT
      COMMON/SLP10 /AIKA(50),IIKA(50),JIKA(50),IKA
      COMMON/SLP11 /BIKB(50),CCIKCC(50),IIKB(50),IIKCC(50),IKB,IKCC
      COMMON/SLP12 /INITIA
      COMMON/SLP14 /KPHASE
      COMMON/ACHECK/IACH
      LOGICAL CB
      EPS=1.0E-4
      ONE  =1.0E0
      PGRE =1.0E30
      DO 10 J=1,N
      C(J)=ZERO
   10 CCC(J)=ZERO
      CALL SET2
      ZINT=Z
    3 CALL PHASE1
      WRITE(6,6999)
 6999 FORMAT(1H1)
      IF(INITIA.EQ.0) GO TO 12
      IF(IACH.LT.1) GO TO 925
      DO 924 I=1,M
      IF(I.GT.M1) GO TO 926
      IK=LBXL(I)
      WRITE(6,693)I,IK
  693 FORMAT(1H0,5X,2I5)
      GO TO 927
  926 II=I-M1
      WRITE(6,694)I,II
  694 FORMAT(1H0,5X,2I5)
  927 WRITE(6,695)(A(I,J),J=1,N0)
  695 FORMAT(1H ,/(5X,10E12.4))
  924 CONTINUE
  925 CONTINUE
      GO TO 11
   12 CONTINUE
      IF(ABS(Z/ZINT)-EPS) 5,5,999
 999  WRITE(6,601)
 601  FORMAT(//1H ,54H NOW WE HAVE FOUND THAT THERE EXISTS NO FEASIBLE P
     1OINT///1H 30(4HEND ))
      RETURN
   11 CONTINUE
    5 IF(KPHASE.EQ.1) GO TO 20
      CALL PHASE2
   20 CONTINUE
      RETURN
```

SOURCE STATEMENT

```
      END
```

SOURCE STATEMENT

```
C     SUBROUTINE
       SUBROUTINE SET2
       COMMON/SLP1  /A(345,355),B(500),CC(500)
       COMMON/SLP2  /M1,M2,M3,NO,NAME(5),JLPW,NITERE
       COMMON/SLP3  /BP(500),C(500),IB(500),NUM(500)
       COMMON/SLP4  /Z,ZERO,ONE,PGRE,EPS
       COMMON/SLP5  /M,N,NC,N1F(11),N2F(5),LS,NAX,MAX
       COMMON/SLP6  /CB(450),CCC(450)
       COMMON/SLP12 /INITIA
       LOGICAL CB
       MM=M1+1
       NOM2=NO+M2
       IF(INITIA.NE,0) GO TO 20
       DO 4 J=1,M
        NOM2J= NOM2+J
         DO 3 I=1,M
          IF(I-J) 1,2 , 1
   1      A(I,NOM2J)=ZERO
          GO TO 3
   2      A(I,NOM2J)=ONE
   3     CONTINUE
   4   CONTINUE
   20  CONTINUE
        IF(M2) 7,7,5
   5   DO 6 J=1,M2
        NOJ=NO+J
        M1J=M1+J
        C(NOJ)=ONE
       A(M1J,NOJ) =-ONE
   6   CONTINUE
   7   DO 8 J=1,N
       CB(J)=.FALSE,
       IF(J.GT.NOM2) CB(J)=.TRUE,
   8   NUM(J)=J
       MMM=M
       IF(INITIA.NE,0) MMM=M1
       DO 9 I=1,MMM
   9   IB(I)= I+NO+M2
       IF(M2+M3) 14,14,10
   10  Z=ZERO
       DO 11 I=MM,M
  11    Z=Z-B(I)
       DO 13 J=1,NO
       C(J)=ZERO
        CCC(J)=-CC(J)
       DO 12 I=MM,M
  12      C(J)=C(J)-A(I,J)
  13   CONTINUE
  14   RETURN
       END
```

SOURCE STATEMENT

```
C     SUBROUTINE
      SUBROUTINE PHASE1
      COMMON/SAXEQB/XI(500,1),IPV
      COMMON/SLP1  /A(345,355),B(500),CC(500)
      COMMON/SLP2  /M1,M2,M3,N0,NAME(5),JLPW,NITERE
      COMMON/SLP3  /BP(500),C(500),IB(500),NUM(500)
C     COMMON/SLP4  /Z,ZERO,ONE,PGRE,EPS
      COMMON/SLP4  /ZOB,ZERO,ONE,PGRE,EPS
      COMMON/SLP5  /M,N,NC,N1F(11),N2F(5),LS,NAX,MAX
      COMMON/SLP6  /CB(450),CCC(450)
      COMMON/SLP7  /XL(200),NXLS(200),NXLE(200),NFUS(200),NFUE(200)
      COMMON/SLP8  /LBXL(200),LBXLE,KFU(200),KFUE
      COMMON/SLP9  /MXLG,NFUS,NFUN,LPDT
      COMMON/SLP10 /AIKA(50),IIKA(50),JIKA(50),IKA
      COMMON/SLP11 /BIKB(50),CCIKCC(50),IIKB(50),IIKCC(50),IKB,IKCC
      COMMON/SLP12 /INITIA
      COMMON/SLP15 /QCL(200),LQ(200),LFUE,LQE,LQQ(200),LIQ(200)
      COMMON/DOM   /IMOJI(20)
      COMMON/D12/JFIG
      COMMON/SAUT/  XSTART,YSTART,XEND,YEND,DMX,DMY,MXE,MYE
      COMMON/DATA68/BC(500,1)
      COMMON/DATA8R/BV(300,1),BL(300),NBN,JBN,NF(300)
      COMMON/SJEXP/JEXP
      COMMON/SPF26 /IPR
      COMMON/ACHECK/IACH
      COMMON/SPA   /MACP,MICP
      COMMON/DATAHF/IHF
      COMMON/FKIND/ KIND,KINDD,KINDK,KINDV
      COMMON/SPART/ICN(20,3),ICNS(20,3),ICNE(20,3),IPTE,KEPE(20)
      COMMON/DATA0R/NF331,NF332,NF333,IEK,INK,IFK,IFD,KNI
      COMMON/DATA1R/IAD,KFEM1,MTITL(5)
      COMMON/DATA2R/Z(500,2),NP
      COMMON/DATA7R/VE(500,2),VN(500,2),IVEL,IVG,IVEX,IVW,IEXP
      COMMON/DATA3C/CSEIDO
      COMMON/DATA6R/ME,JQD
      COMMON/DATA3R/AREAS,SEIDO,NE,INKE,IAREA,ISEIDO,NOD(500,6)
      COMMON/DATA8C/NNFF(500)
      COMMON/DATA9R/KEISAN,IAB,NAB(500)
      COMMON/SJAXB /JAXB
      COMMON/SSKMT/HH(6,6)
      COMMON/RESULT/X(500)
C     COMMON/SUNIB /B(500,1)
      COMMON/SUNIB /B0(500,1)
      COMMON/SBCHEK/IBCHEK(20) ,IBCH
      COMMON/ACRACA/ACR(20,50),ACA(5,120)
      LOGICAL CB
      IPHASE=1
      WRITE(6,600) IPHASE,M,N
  600 FORMAT(1H //1X,5HPHASE,I1,5H   M=I3,5H,   N=I3//)
      IF(KEISAN.EQ.0) GO TO 100
      IF(INITIA) 199,201,199
  199 CONTINUE
      DO 2 I=1,NP
      DO 2 J=1,NP
    2 A(I,J)=0.0
      DO 8 KM=1,NE
```

SOURCE STATEMENT

```
      CALL SKMT(KM)
      DO 1 I=1,KNI
      II=NOD(KM,I)
      DO 1 J=1,KNI
      JJ=NOD(KM,J)
    1 A(II,JJ)=A(II,JJ)+HH(I,J)
    8 CONTINUE
      DO 723 I=1,NP
      DO 723 MM=1,ME
  723 BC(I,MM)=BO(I,MM)
 8001 DO 801 I=1,NP
  801 NNFF(I)=0
      IF(NBN.EQ.0) GO TO 901
      DO 802 K=1,NBN
      KF=NF(K)
      NNFF(KF)=1
      DO 803 MM=1,ME
  803 BO(KF,MM)=BV(K,MM)
  802 CONTINUE
  901 CONTINUE
      DO 25 I=1,NP
      DO 25 MM=1,ME
   25 XI(I,MM)=BO(I,MM)
CC BCHECKS
      IF(IBCH.EQ.0) GO TO 292
      DO 238 I=1,IBCH
      MP=IBCHEK(I)
      DO 238 J=1,NP
  238 ACA(I,J)=A(MP,J)
CC    BCHECKE
  292 CONTINUE
      IF(NBN.EQ.0) GO TO 211
      DO 210 I=1,NBN
      KF=NF(I)
      MKF=KF
CC    BRCHECKS
      IF(KEISAN.LT.2) GO TO 71
      DO 70 J=1,NP
   70 ACR(I,J)=A(MKF,J)
   71 CONTINUE
CC    BRCHEKE
      DO 210 J=1,NO
      A(MKF,J)=0.0
      IF(KF.EQ.J) A(MKF,J)=1.0
  210 CONTINUE
  211 CONTINUE
      CALL PAXEQB(NP,ME,NO,M1,    M3,M,NFUN)
      WRITE(6,618)
  618 FORMAT(1H1,20X,'RESULT ',///)
      DO 20 I=1,NP
      WRITE(6,601) I,(Z(I,J),J=1,IFD),(XI(I,MM),MM=1,ME)
  601 FORMAT(1H ,5X,I5,2F10.2,5E12.5)
   20 CONTINUE
CCC   FIGURE START
      IF(JFIG.EQ.0) GO TO 1283
CHUI S.AUT NO KXE,KYE DEWANAI
```

SOURCE STATEMENT

```
      IF(KNI.EQ.3) GO TO 9025
      KXE=MXE*2+1
      KYE=MYE*2+1
      GO TO 9026
 9025 CONTINUE
      KXE=MXE+1
      KYE=MYE+1
 9026 CONTINUE
      WRITE(6,650)IMOJI
  650 FORMAT(1H1,5(/),19X,32(3H*  ),//19X,1H*,92X,1H*,//19X,1H*,5X,
     1        20A4,7X,1H*,//19X,1H*,92X,1H*,//19X,32(3H*  ),5(/))
      WRITE(6,6620)KXE,KYE
 6620 FORMAT(1H0,/,26X,'KXE =',I3,4X,'KYE =',I3,/)
      IF(KXE.GT.21) GO TO 1288
      DO 1290 MM=1,ME
      WRITE(6,6660)
 6660 FORMAT(1H1,//)
      DO 1281 I=1,KYE
       IF(JFIG) 1282,1283,1284
 1282 NS=(I-1)*KXE+1
      NH=I*KXE
      IY=I
      GO TO 1285
 1284 NS=NP-I*KXE+1
      NH=NP-(I-1)*KXE
      IY=KYE-I+1
 1285 WRITE(6,6661)IY,(XI(IN,MM),IN=NS,NH)
 6661 FORMAT(1H0,//,I3,1X,21F6.1)
 1281 CONTINUE
 1290  CONTINUE
      GO TO 1283
 1288 WRITE(6,6662)
 6662 FORMAT(1H0,//,20X,'KXE.GT.21 DE FIGURE NASI',/)
 1283 CONTINUE
CCC   FIGURE END
      DO 15 I=1,NP
      DO 16 MM=1,ME
   16 BO(I,MM)=BC(I,MM)
   15 CONTINUE
      IF(INITIA.GE.1) GO TO 203
  200 DO 202 I=1,NP
  202 XI(I,1)=0.0
  203 CONTINUE
      DO 7050 L=1,LBXLE
      LL=L+N0
      IB(L)=LL
      KK=LBXL(L)
 7050 B(L)=B(L)-XI(KK,1)
      DO 7051 NL=1,M3
      MNL=M1+NL
      IB(MNL)=NL
 7051 B(MNL)=XI(NL,1)
      DL 7052 II=1,NFUN
      LL=M3+II
      MII=M+II
      IB(MII)=LL
```

SOURCE STATEMENT

```
 7052 B(MII)=0.0
      IF(IACH.LT.1) GO TO 8070
      WRITE(6,6807)
 6807 FORMAT(1H ,12X,'* I,     B(I) *',//)
      DO 8071 I=1,N
      WRITE(6,6808)I,B(I)
 6808 FORMAT(1H ,10X,I5,E12.4)
 8071 CONTINUE
 8070 CONTINUE
      GO TO 204
  201 CALL PIV
  204 CONTINUE
  100 CONTINUE
      RETURN
      END
```

SOURCE STATEMENT

```
C     SUBROUTINE
      SUBROUTINE PAXEQB(N,ME,N0,M1,    M3,M,NFUN)
C     COMMON/SLP1  /A(360,380),B(500),CC(500)
      COMMON/SLP1  /A(345,355),BB(500),CC(500)
      COMMON/SLP3  /BP(500),C(500),IB(500),NUM(500)
      COMMON/SLP4  /Z,ZERO,ONE,PGRE,EPS
      COMMON/SLP6  /CB(450),CCC(450)
      COMMON/SLP8  /LBXL(200),LBXLE,KFU(200),KFUE
      COMMON/SLP12 /INITIA
      COMMON/SLP15 /QCL(200),LQ(200),LFUE,LQE,LQQ(200),LIQ(200)
      COMMON/SAXEQB/B(500,1),IPV
      COMMON/SJAXB /JAXB
      COMMON/FEMFD/JFEMFD
      COMMON/ACHECK/IACH
      DIMENSION TA(500)
      DIMENSION BAX(500)
      DIMENSION R(500)
      LOGICAL CB
      NP=N
      N0M3=N0+M1
      IPV=1
      DO 401 IN=1,N
      DO 401 IK=1,NFUN
      KIK=N+IK
  401 A(IN,KIK)=0.0
      DO 402 IK=1,NFUN
      KIK=N+IK
      KK=KFU(IK)
      IF(JFEMFD.GT.0) GO TO 408
      A(KK,KIK)=-1.0
      GO TO 402
  408 A(KK,KIK)=1.0
  402 CONTINUE
      IF(N.NE.1) GO TO 75
      DO 76 MM=1,ME
   76 B(1,MM)=B(1,MM)/A(1,1)
      GO TO 28
   75 CONTINUE
      NM1=N-1
      DO 1 K=1,NM1
      IF(IACH.NE.3) GO TO 30
      IF(INITIA.EQ.0) GO TO 30
      WRITE(6,630) K
  630 FORMAT(1H1,//10X,'K =',I4,10X,'SIMPLEX TABLEAU',//)
      IF(N0M3.GT.15) GO TO 120
      WRITE(6,635)(CC(I),I=1,N0M3)
  635 FORMAT(1H ,22X,15F7.3)
      WRITE(6,636)
  636 FORMAT(1H0,' I   B.V.     C     B.F.S. X( 1)  X( 2)  X( 3)  X( 4)',
     1'  X( 5)  X( 6)  X( 7)  X( 8)  X( 9)  X(10)  X(11)  X(12)  ',
     2  'X(13)  X(14)  X(15)',/)
  120 CONTINUE
      DO 31 L=1,M3
      IF(N0M3.GT.15) GO TO 32
      IIB=M1+M2+L
      DO 90 I=1,M1
```

SOURCE STATEMENT

```
   90 R(I)=0.0
      WRITE(6,631)IIB,L,CC(L),B(L,1),(A(L,J),J=1,N0),(R(I),I=1,M1)
  631 FORMAT(1H0,I2,' X(',I2,') ',F6.3,F7.3,15F7.3)
      GO TO 31
   32 CONTINUE
      WRITE(6,632)IIB,L,B(L,1)
  632 FORMAT(1H ,2I5,E10.2)
      WRITE(6,633)(A(L,J),J=1,N0M3)
  633 FORMAT(1H ,/(5X,10E9.2))
   31 CONTINUE
      DO 81 I=1,M1
      N0I=N0+I
      IF(N.GT.15) GO TO 82
      IBI=IB(I)
      IF(LFUE.NE.0.AND.I.GT.LBXLE) GO TO 200
      LI=LBXL(I)
      GO TO 201
  200 IL=I-LBXLE
      LI=NP+LIQ(IL)
  201 CONTINUE
      DO 800 J=1,N0M3
      R(J)=0.0
      IF(N0I.EQ.J) R(J)=1.0
      IF(LI.EQ.J) R(J)=1.0
  800 CONTINUE
      IBI=IB(I)
      WRITE(6,631)I,IB(I),CC(IBI),BB(I),(R(J),J=1,N0M3)
      GO TO 81
   82 CONTINUE
      WRITE(6,632)I,IB(I),BB(I)
      WRITE(6,633)(R(J),J=1,N0M3)
      GO TO 300
   81 CONTINUE
      WRITE(6,637)Z,(CCC(J),J=1,N0M3)
  637 FORMAT(1H0,5X,'Z',9X,F7.3,15F7.3)
  300 CONTINUE
      M3P1=M3+1
      N0P1=N0+1
      WRITE(6,680)
  680 FORMAT(1H0,//,5X,'B.V. ; BASIC VARIABLE',/,
     1  5X,'  C  ; COEFFICIENT OF OBJECTIVE FUNCTION',/,  3X,
     2   'B.F.S. ; BASIC FEASIBLE SOLUTION',/)
      WRITE(6,681)M3,M3P1,N0,N0P1,N0M3
  681 FORMAT(1H0,2X,'X( 1) ~ X(',I2,') ; STATE VARIABLE',/,
     1               3X,'X(',I2,') ~ X(',I2,') ; DECISION VARIABLE',/,
     2               3X,'X(',I2,') ~ X(',I2,') ; SLACK VARIABLE',/)
   30 CONTINUE
      K1=K+1
C     PIVOT WO 1 NI SURU
      AKK=A(K,K)
      DO 6 KJ=K,N
      A(K,KJ)=A(K,KJ)/AKK
    6 CONTINUE
      DO 400 IK=1,NFUN
      KIK=N+IK
  400 A(K,KIK)=A(K,KIK)/AKK
```

SOURCE STATEMENT

```
      DO 41 MM=1,ME
   41 B(K,MM)=B(K,MM)/AKK
C     ATARASII KEISU WO MOTOMERU
      DO 7 I=K1,N
      AIK=A(I,K)
      DO 42 MM=1,ME
   42 B(I,MM)=B(I,MM)-AIK*B(K,MM)
      DO 403 IK=1,NFUN
      KIK=N+IK
  403 A(I,KIK)=A(I,KIK)-AIK*A(K,KIK)
      DO 7 J=K,N
      A(I,J)=A(I,J)-AIK*A(K,J)
    7 CONTINUE
    1 CONTINUE
C     KOHAN
      DO 43 MM=1,ME
   43 B(N,MM)=B(N,MM)/A(N,N)
      DO 404 IK=1,NFUN
      KIK=N+IK
  404 A(N,KIK)=A(N,KIK)/A(N,N)
      DO 8 II=1,NM1
      I=N-II
      I1=I+1
      DO 44 MM=1,ME
   44 BAX(MM)=B(I,MM)
      DO 405 IK=1,NFUN
      KIK=N+IK
  405 TA(IK)=A(I,KIK)
      DO 9 J=I1,N
      DO 45 MM=1,ME
   45 BAX(MM)=BAX(MM)-A(I,J)*B(J,MM)
      DO 406 IK=1,NFUN
      KIK=N+IK
  406 TA(IK)=TA(IK)-A(I,J)*A(J,KIK)
    9 CONTINUE
      DO 46 MM=1,ME
   46 B(I,MM)=BAX(MM)
      DO 407 IK=1,NFUN
      KIK=N+IK
  407 A(I,KIK)=TA(IK)
    8 CONTINUE
      NOM1=NO+M1
      DO 495 I=1,M3
      M3I1=M3-I+1
      DO 496 J=1,NOM1
      MI1=M-I+1
  496 A(MI1,J)=A(M3I1,J)
  495 CONTINUE
      DO 480 I=1,M1
      DO 481 J=1,NOM1
      A(I,J)=0.0
      JNO=J-NO
      IF(JNO.EQ.I) A(I,J)=1.0
  481 CONTINUE
      IF(LFUE.NE.0.AND.I.GT.LBXLE) GO TO 220
      DO 482 J=1,NFUN
```

SOURCE STATEMENT

```
      LL=LBXL(I)
      NJ=N+J
      ML=M1+LL
  482 A(I,NJ)=-A(ML,NJ)
      GO TO 480
  220 IL=I-LBXLE
      LI=NP+LIQ(IL)
      A(I,LI)=1.0
  480 CONTINUE
      DO 485 I=1,NOM1
      CB(I)=.TRUE.
      IF(I.GT.N.AND.I.LE.NO) CB(I)=.FALSE.
  485 CONTINUE
      IF(IACH.LT.1) GO TO 925
      DO 924 I=1,M
      IF(I.GT.M1) GO TO 926
      IK=LBXL(I)
      WRITE(6,693)I,IK
  693 FORMAT(1H0,5X,2I5)
      GO TO 927
  926 II=I-M1
      WRITE(6,694)I,II
  694 FORMAT(1H0,5X,2I5)
  927 WRITE(6,695)(A(I,J),J=1,NO)
  695 FORMAT(1H ,/(5X,10E12.4))
  924 CONTINUE
  925 CONTINUE
      M11=M1+1
      DO 700 I=M11,M
      I1=I-M1
      DO 700 J=1,N
      A(I,J)=0.0
      IF(I1.EQ.J) A(I,J)=1.0
  700 CONTINUE
   28 RETURN
      END
```

SOURCE STATEMENT

```
C      SUBROUTINE
       SUBROUTINE PHASE2
       COMMON/SLP1  /A(345,355),B(500),CC(500)
       COMMON/SLP2  /M1,M2,M3,N0,NAME(5),JLPW,NITERE
       COMMON/SLP3  /BP(500),C(500),IB(500),NUM(500)
       COMMON/SLP4  /Z,ZERO,ONE,PSRE,EPS
       COMMON/SLP5  /M,N,NC,N1F(11),N2F(5),LS,NAX,MAX
       COMMON/SLP6  /CB(450),CCC(450)
       LOGICAL CB
       N=N0+M2+M1
       IPHASE =2
       WRITE(6,600) IPHASE,M,N
  600  FORMAT(1H //1X,5HPHASE,I1,5H    M=I3,5H,   N=I3//)
       DO 1 J=1,N0
    1  C(J)= -CC(J)
       NN0=N0+1
       DO 2 J=NN0,N
    2  C(J)=ZERO
       Z=ZERO
       DO 4 I=1,M
        K=IB(I)
        CM=C(K)
        DO 3 J=1,N
       IF(CB(J)) GO TO 3
        C(J)= C(J)-A(I,J)*CM
    3  CONTINUE
       Z=Z-CM*B(I)
    4  CONTINUE
       DO 5 J=1,N
          IF(CB(J)) C(J)=ZERO
    5  CONTINUE
       CALL PIV
       WRITE(6,601)
  601  FORMAT(//1H     ,30(4HEND ) )
       RETURN
       END
```

SOURCE STATEMENT

```
C     SUBROUTINE
      SUBROUTINE PIV
      COMMON/SLP1  /A(345,355),B(500),CC(500)
      COMMON/SLP2  /M1,M2,M3,NO,NAME(5),JLPW,NITERE
      COMMON/SLP3  /BP(500),C(500),IB(500),NUM(500)
      COMMON/SLP4  /Z,ZERO,ONE,PGRE,EPS
      COMMON/SLP5  /M,N,NC,N1F(11),N2F(5),LS,NAX,MAX
      COMMON/SLP6  /CB(450),CCC(450)
      COMMON/SLP12 /INITIA
      COMMON/SLP13 /CUT
      COMMON/ACHECK/IACH
      LOGICAL CB
      WRITE(6,600)
 600  FORMAT(//1H ,33HTHE INITIAL TABLEAU IS AS FOLLOWS//)
      CALL PRINT
C *** SEARCHING OF PIVOTAL COLUMN ***
      LAMP=0
      ITERE=0
    1 JC=0
      IF(IACH.NE.3) GO TO 30
      IF(INITIA.EQ.0) GO TO 30
      WRITE(6,630)ITERE
  630 FORMAT(1H1,//,10X,'ITERE =',I4,   10X,'SIMPLEX TABLEAU',//)
      IF(N.GT.15) GO TO 120
      WRITE(6,635)(CC(I),I=1,N)
  635 FORMAT(1H ,22X,15F7.3)
      WRITE(6,636)
  636 FORMAT(1H0,' I  B.V.     C    B.F.S. X( 1)  X( 2)  X( 3)  X( 4)',
     1'  X( 5)  X( 6)  X( 7)  X( 8)  X( 9)  X(10)  X(11)  X(12)  ',
     2  'X(13)  X(14)  X(15)',/)
  120 CONTINUE
      M11=M1+1
      DO 31 I=M11,M
      IF(N.GT.15) GO TO 32
      IBI=IB(I)
      WRITE(6,631)I,IB(I),CC(IBI),B(I),(A(I,J),J=1,N)
  631 FORMAT(1H0,I2,' X(',I2,') ',F6.3,F7.3,15F7.3)
      GO TO 31
   32 CONTINUE
      WRITE(6,632)I,IB(I),B(I)
  632 FORMAT(1H ,2I5,E10.2)
      WRITE(6,633)(A(I,J),J=1,N)
  633 FORMAT(1H ,/(5X,10E9.2))
   31 CONTINUE
      DO 81 I=1,M1
      IF(N.GT.15) GO TO 82
      IBI=IB(I)
      WRITE(6,631)I,IB(I),CC(IBI),B(I),(A(I,J),J=1,N)
      GO TO 81
   82 CONTINUE
      WRITE(6,632)I,IB(I),B(I)
      WRITE(6,633)(A(I,J),J=1,N)
      GO TO 300
   81 CONTINUE
      WRITE(6,637)Z,(C(J),J=1,N)
  637 FORMAT(1H0,5X,'Z',9X,F7.3,15F7.3)
```

SOURCE STATEMENT

```
  300 CONTINUE
      M3P1=M3+1
      NOP1=NO+1
      WRITE(6,680)
  680 FORMAT(1H0,//,5X,'B.V. ; BASIC VARIABLE',/,
     1  5X,'  C  ; COEFFICIENT OF OBJECTIVE FUNCTION',/,  3X,
     2   'B.F.S. ; BASIC FEASIBLE SOLUTION',/)
      WRITE(6,681)M3,M3P1,NO,NOP1,N
  681 FORMAT(1H0,2X,'X( 1) ~ X(',I2,') ; STATE VARIABLE',/,
     1              3X,'X(',I2,') ~ X(',I2,') ; DECISION VARIABLE',/,
     2              3X,'X(',I2,') ~ X(',I2,') ; SLACK VARIABLE',/)
   30 CONTINUE
      ITERE=ITERE+1
      CNEW =ZERO
      DO 17 J=1,N
         T=C(J)
         IF(T.GE.CUT) GO TO 17
         IF(LAMP.NE.0) GO TO 950
         IF(CNEW.GE.ABS(T)) GO TO 17
         CNEW = ABS(T)
         JC=J
  17  CONTINUE
      IF(JC.NE.0) GO TO 26
      GO TO 900
  26  CONTINUE
      CB(JC)=.TRUE.
C *** SEARCHING OF PIVOTAL ROW ***
      GRE =PGRE
      IR=0
      DO 18 I=1,M
         T=A(I,JC)
         IF(T.LT.EPS) GO TO 18
         WS= B(I)/T
         IF(WS.GT.GRE) GO TO 18
         GRE = WS
         IR=I
  18  CONTINUE
      IF(IR.EQ.0) GO TO 950
      K=IB(IR)
      CB(K)=.FALSE.
C *** SWEEPING ***
      P=A(IR,JC)
      T=C(JC)
      Z=Z-T*B(IR)/P
      DO 19 J=1,N
         C(J)=C(J)-A(IR,J)/P*T
  19  CONTINUE
      B(IR)=B(IR)/P
      DO 20 I=1,M
         IF(I.EQ.IR) GO TO 20
         B(I)=B(I)-B(IR)*A(I,JC)
  20  CONTINUE
      DO 21 J=1,N
         IF(J.EQ.JC) GO TO 21
         A(IR,J)=A(IR,J)/P
  21  CONTINUE
```

SOURCE STATEMENT

```
      A(IR,JC)=ONE
      DO 22 J=1,N
         IF(J,EQ.JC) GO TO 22
       DO 23 I=1,M
          IF(I.EQ.IR) GO TO 23
          A(I,J)=A(I,J)-A(IR,J)*A(I,JC)
   23  CONTINUE
   22 CONTINUE
      DO 25 I=1,M
         IF(I,EQ.IR) GO TO 25
         A(I,JC)=ZERO
   25 CONTINUE
      IB(IR)=JC
      MINT=MOD(ITERE,NITERE)
      IF(MINT.NE.0) GO TO 70
      WRITE(6,620)ITERE
      CALL PRINT
    70 CONTINUE
      GO TO 1
C *** IN CASE THERE IS THE OPTIMAL VALUE
  900 WRITE(6,601)
  601 FORMAT(////40H  NOW WE HAVE REACHED THE OPTIMAL VALUE )
      WRITE(6,620)ITERE
   620 FORMAT(1H1,//,5X,'ITERE =',I5,/)
      CALL PRINT
      RETURN
C *** IN CASE THERE IS NO BOUND
  950 WRITE(6,602)
  602 FORMAT(////48H THE VALUE OF OBJECTIVE FUNCTION CAN BE INFINITE   )
      WRITE(6,620)ITERE
      CALL PRINT
      RETURN
      END
```

SOURCE STATEMENT

```
C     SUBROUTINE
      SUBROUTINE PREPRI
      COMMON/SLP1  /A(345,355),B(500),CC(500)
      COMMON/SLP2  /M1,M2,M3,N0,NAME(5),JLPW,NITERE
      COMMON/SLP3  /BP(500),C(500),IB(500),NUM(500)
      COMMON/SLP4  /Z,ZERO,ONE,PGRE,EPS
      COMMON/SLP5  /M,N,NC,N1F(11),N2F(5),LS,NAX,MAX
      DIMENSION NN1F(11),NN2F(5)
      DATA NN1F/'(1H ////    ,2X,0I20/(1H ,6HZJ-CJ= ,0E20.7)) '/
      DATA NN2F/'(1H ,5X,I3,0E20.7)  '/
      LLS=N-LS
      DO 100 I=1,11
      N1F(I)=NN1F(I)
  100 CONTINUE
      DO 200 I=1,5
      N2F(I)=NN2F(I)
  200 CONTINUE
      N1F(4)=N1F(4)+LLS
      N1F(9)=N1F(9)+LLS
      N2F(3)=N2F(3)+LLS
      RETURN
      END
```

SOURCE STATEMENT

```
C      SUBROUTINE
       SUBROUTINE PRINT
       COMMON/SLP1  /A(345,355),B(500),CC(500)
       COMMON/SLP2  /M1,M2,M3,N0,NAME(5),JLPW,NITERE
       COMMON/SLP3  /BP(500),C(500),IB(500),NUM(500)
       COMMON/SLP4  /Z,ZERO,ONE,PGRE,EPS
       COMMON/SLP5  /M,N,NC,N1F(11),N2F(5),LS,NAX,MAX
       COMMON/SLP6  /CB(450),CCC(450)
       LOGICAL CB
       WRITE(6,600) Z
  600  FORMAT(1H //5X,33HTHE VALUE OF OBJECTIVE FUNCTION =,E16.7///)
       DO 10 I=1,M
   10  WRITE(6,601) IB(I),B(I)
  601  FORMAT(1H 10X,2HXB,I3,2H =E16.7)
       IF(JLPW.NE.999) GO TO 15
       L=N/5
       LS=L*5
       IF(N.LT.5) GO TO 288
       LLS=LS-4
       DO 11 K=1,LLS,5
       KK=K+4
       WRITE(6,602) (NUM(J),J=K,KK),(C(J),J=K,KK),(CCC(J),J=K,KK)
   602 FORMAT(///1H ,2X,5I20/(1H ,2X,6HZJ-CJ= ,5E20.7))
         DO 12 I=1,M
    12 WRITE(6,603) IB(I),(A(I,J),J=K,KK)
  603    FORMAT(1H ,5X,I3,5E20.7)
        WRITE(6,604)
  604   FORMAT(///1H ,24(5H* * *))
   11  CONTINUE
       IF(N.EQ.LS) GO TO 14
  288  CALL PREPRI
       LLLS=LS+1
       WRITE(6,N1F) (NUM(J),J=LLLS,N),(C(J),J=LLLS,N),(CCC(J),J=LLLS,N)
       DO 13 I=1,M
    13 WRITE(6,N2F) IB(I),(A(I,J),J=LLLS,N)
       WRITE(6,604)
    14 CONTINUE
    15 CONTINUE
       RETURN
       END
```

SOURCE STATEMENT

```
C     SUBROUTINE
      SUBROUTINE LPREAD(MOJI,N?)
      COMMON/SLP2  /M1,M2,M3,N0,NAME(5),JLPW,NITERE
      COMMON/SLP3  /BP(500),C(500),IB(500),NUM(500)
      COMMON/SLP4  /Z,ZERO,ONE,PGRE,EPS
      COMMON/SLP5  /M,N,NC,N1F(11),N2F(5),LS,NAX,MAX
      COMMON/SLP6  /CB(450),CCC(450)
      COMMON/SLP7  /XL(200),NXLS(200),NXLE(200),NFUS(200),NFUE(200)
      COMMON/SLP8  /LBXL(200),LBXLE,KFU(200),KFUE
      COMMON/SLP9  /MXLG,NFUG,NFUN,LPDT
      COMMON/SLP10 /AIKA(50),IIKA(50),JIKA(50),IKA
      COMMON/SLP11 /BIKB(50),CCIKCC(50),IIKB(50),IIKCC(50),IKB,IKCC
      COMMON/SLP12 /INITIA
      COMMON/SLP15 /QCL(200),LQ(200),LFUE,LQF,LQQ(200),LIQ(200)
      COMMON/SLP13 /CUT
      COMMON/SLP14 /KPHASE
      COMMON/D2R   /XSTART,YSTART,XEND,YEND,DMX,DMY,MXE,MYE,KXE,KYE
      COMMON/D2CC  /SAREA
      COMMON/DATA9R/KEISAN,IAB,NAB(500)
      COMMON/FEMFD/JFEMFD
      DIMENSION MOJI(5)
      LOGICAL CB
      MAX=345
      NAX=355
      CUT=-0.1E-4
      DO 1119 I=1,5
 1119 NAME(I)=MOJI(I)
      LPDT=0
  333 READ(5,570)ID,LD,LD1,LD2
  570 FORMAT(I2,2X,I1,2I5)
      IF(ID.NE.11) GO TO 399
      GO TO(91,92,93,94,95,96,97),LD
   91 WRITE(6,670)LD
  670 FORMAT(1H0,//,5X,'*  11 - ',I1,' *',//)
      MXLG=LD1
      WRITE(6,631)MXLG
  631 FORMAT(1H0,3X,'MXLG =',I5,//)
      IF(MXLG.EQ.0) GO TO 333
      READ(5,550)(NXLS(I),NXLE(I),XL(I),I=1,MXLG)
  550 FORMAT(4(2I5,F10.0))
      WRITE(6,660)(I,NXLS(I),NXLE(I),XL(I),I=1,MXLG)
  660 FORMAT(1H ,/(3X,4(3I5,F12.4)))
      GO TO 333
   92 WRITE(6,670) LD
      NFUG=LD1
      LFUE=LD2
      JCD=ID3
      WRITE(6,632)NFUG,LFUE,JCD
  632 FORMAT(1H0,/,3X,'NFUG =',I5,5X,'LFUE =',I5,5X,'JCD =',I5,//)
      IF(NFUG.EQ.0.AND.LFUE.EQ.0) GO TO 333
      IF(LFUE.NE.0) GO TO 3000
      READ(5,551)(NFUS(I),NFUE(I),I=1,NFUG)
  551 FORMAT(16I5)
      WRITE(6,661)(I,NFUS(I),NFUE(I),I=1,NFUG)
  661 FORMAT(1H ,/(3X,8(3I5)))
      GO TO 333
```

SOURCE STATEMENT

```
3000 READ(5,500)(KFU(I),LQ(I),QCL(I),I=1,LFUE)
 500 FORMAT(4(2I5,F10.0))
     WRITE(6,600)(I,KFU(I),LQ(I),QCL(I),I=1,LFUE)
 600 FORMAT(1H ,/(3X,4(3I5,F12.4)))
     IF(JFEMFD.NE.1) GO TO 921
     IF(JCD.NE.0) GO TO 921
     NK1=NP-KXE+1
     DO 920 I=1,LFUE
     IF(LQ(I).EQ.0) GO TO 920
     II=KFU(I)
     MKC=MOD(II,KXE)
     IF(II.LE.KXE.OR.II.GE.NK1) GO TO 922
     IF(MKC.EQ.1.OR.MKC.EQ.0) GO TO 922
     QCL(I)=QCL(I)/SAREA
     GO TO 920
 922 IF(II.EQ.1.OR.II.EQ.KXE.OR.II.EQ.NK1.OR.II.EQ.NP) GO TO 923
     QCL(I)=QCL(I)/((SAREA/2.0))
     GO TO 920
 923 QCL(I)=QCL(I)/((SAREA/4.0))
 920 CONTINUE
 921 CONTINUE
     LQE=0
     DO 3005 I=1,LFUE
     IF(LQ(I).EQ.0) GO TO 3005
     LQE=LQE+1
     LQQ(LQE)=KFU(I)
     LIQ(LQE)=I
3005 CONTINUE
     GO TO 333
  93 WRITE(6,670) LD
     IKA=LD1
     WRITE(6,633)IKA
 633 FORMAT(1H0,/3X,'IKA =',I5,//)
     IF(IKA.EQ.0) GO TO 333
     READ(5,554)(IIKA(I),JIKA(I),AIKA(I),I=1,IKA)
 554 FORMAT(4(2I5,F10.0))
     WRITE(6,640)(I,IIKA(I),JIKA(I),AIKA(I),I=1,IKA)
 640 FORMAT(1H ,/(3X,4(3I5,F12.4)))
     GO TO 333
  94 WRITE(6,670)LD
     IKB=LD1
     WRITE(6,634)IKB
 634 FORMAT(1H0,/,3X,'IKB =',I5,//)
     IF(IKB.EQ.0) GO TO 333
     READ(6,555)(IIKB(I),BIKB(I),I=1,IKB)
 555 FORMAT(4(I5,5X,F10.0))
     WRITE(6,641)(I,IIKB(I),BIKB(I),I=1,IKB)
 641 FORMAT(1H0,/(3X,4(I5,5X,I5,F12.4)))
     GO TO 333
  95 WRITE(6,670)LD
     IKCC=LD1
     WRITE(6,635)IKCC
 635 FORMAT(1H0,/,3X,'IKCC =',I5,//)
     IF(IKCC.EQ.0) GO TO 333
     READ(6,556)(IIKCC(I),CCIKCC(I),I=1,IKCC)
 556 FORMAT(4(5X,I5,F10.0))
```

SOURCE STATEMENT

```
      WRITE(6,642)(I,IIKCC(I),CCIKCC(I),I=1,IKCC)
  642 FORMAT(1H0,/(3X,4(I5,5X,I5,F12.4)))
      GO TO 333
   96 WRITE(6,670)LD
      JLPW=LD1
      NITERE=LD2
      WRITE(6,680)JLPW,NITERE
  680 FORMAT(1H0,/,3X,'JLPW =',I5,5X,'NITERE =',I5,/)
      GO TO 333
   97 WRITE(6,670)LD
      INITIA=LD1
      KPHASE=LD2
      WRITE(6,675)INITIA,KPHASE
  675 FORMAT(1H0,/3X,'INITIA =',I5,5X,'KPHASE =',I5,/)
      WRITE(6,671)
  671 FORMAT(1H0,//5X,'* LP DATA FINISH *',/)
      M1=0
      DO 120 I=1,MXLG
  120 M1=M1+NXLE(I)-NXLS(I)+1
      LBXLE=M1
      IF(LFUE.NE.0) GO TO 3001
      NFUN=0
      DO 121 I=1,NFUG
  121 NFUN=NFUN+NFUE(I)-NFJS(I)+1
      GO TO 3002
 3001 NFUN=LFUE
      M1=M1+LQE
      WRITE(6,610)M1,LBXLE,LQE
  610 FORMAT(1H0,5X,'(M1=LBXLE+LQE) =',I5,5X,'LBXLE =',I5,5X,'LQE =',I5,
     1/)
 3002 CONTINUE
      M2=0
      M3=NP
      NO=NP+NFUN
      M=M1+M2+M3
      N=NO+M+M2
      IF(INITIA.NE.0) N=NO+M1+M2
      WRITE(6,602)M,N,M1,M2,M3,NO,NFUN
  602 FORMAT(1H0,5X,'(M=M1+M2+M3) =',I5,5X,'(N=NO+M+M2) =',I5
     1     ,5X,'M1 =',I5,5X,'M2 =',I5,5X,'M3 =',I5,5X,'NO =',I5,
     2       5X,'NFUN =',I5,/)
      IF(N-NAX) 4,4,5
    4 IF(M-MAX) 6,6,5
    6 CONTINUE
      GO TO 395
  399 WRITE(6,695)ID,LD
  695 FORMAT(1H0,///,5X,'* LP DATA OKASII *',5X,'ID =',I5,5X,'LD =',
     1     I5,/)
      LPDT=1
      GO TO 395
    5 WRITE(6,658)M,N,MAX,NAX,NAME
  658 FORMAT(1H0,///,5X,'* LP DATA OKASII (M.GT.MAX.OR.N.GT.NAX) *',
     1      'M =',I5,5X,'N =',I5,5X,'MAX =',I5,5X,'NAX =',I5,5X,5A4,/)
      LPDT=1
  395 CONTINUE
      RETURN
      END
```

B. Computer Program for Analytical Method by Double Fourier Series

SOURCE STATEMENT

```
      IMPLICIT  REAL*8 (A-H,O-Z)
      COMMON/DOM    /IMOJI(20)
      COMMON/DOR    /IFK,IFD
      COMMON/FEMFD/JFEMFD
      JFEMFD=1
      IFD=2
      READ(5,520) ICASE
  520 FORMAT(I5)
      WRITE(6,620)ICASE
  620 FORMAT(1H1,//20X,'ICASE =',I5,//)
      DO 10 NCASE=1,ICASE
      WRITE(6,622)NCASE
  622 FORMAT(1H1,//,20X,'*** NCASE =',I5,' ***',10(/))
      WRITE(6,6161)
 6161 FORMAT(1H0,30X,'DOUBLE FOURIER SERIES',/)
      READ(5,505)IMOJI
  505 FORMAT(20A4)
      WRITE(6,650)IMOJI
  650 FORMAT(1H0,5(/),19X,32(3H*  ),//19X,1H*,92X,1H*,//19X,1H*,5X,
     1       20A4,7X,1H*,//19X,1H*,92X,1H*,//19X,32(3H*  ),5(/))
      READ(5,521)IFK
  521 FORMAT(5X,I5)
      WRITE(6,621)IFK
  621 FORMAT(1H1,2X,' * 0 *',//3X,'IFK =',I5,/)
      CALL DFS
   10 CONTINUE
      WRITE(6,600)
  600 FORMAT(1H1)
      STOP
      END
```

```
SOURCE STATEMENT

C      SUBROUTINE
        SUBROUTINE DFS
        IMPLICIT  REAL*8 (A-H,O-Z)
        COMMON/DOM    /IMOJI(20)
        COMMON/D0R    /IFK,IFD
        COMMON/D1R    /KINDU,KINDV,KINDDX,KINDDY,KINDK
        COMMON/D2R    /XSTART,YSTART,XEND,YEND,DMX,DMY,MXE,MYE,KXE,KYE
        COMMON/D2C    /Z(500,2),AREA,SAREA,NP
        COMMON/D3R    /U,V,DX,DY,DK
        COMMON/D7R    /XLE,YHE,JXYE
        COMMON/DR8    /JKE
        COMMON/D12R   /IPR
        COMMON/D12    /JFIG
        COMMON/RESULT/X(500)
        COMMON/MNEND /MEND,NEND,MEN,NEN
        COMMON/FUKA   /XS(50),YS(50),XE(50),YE(50),CQ(50),KCE
        COMMON/PQMN   /FM,FN,P,PML,PNH,QMN
        COMMON/JSIDO /JSD,JQ,JP
        DIMENSION QC(500)
        DIMENSION MS(9),ME(9),KS(9),KE(9)
        DIMENSION SPX(50),SPY(50)
        WRITE(6,6162)
 6162 FORMAT(1H0,30X,'DOUBLE FOURIER SERIES',/)
         CALL DATA
        IF(JXYE.NE.0) GO TO 909
        XLE=XEND-XSTART
        YHE=YEND-YSTART
  909 CONTINUE
        P=3.141592654
        WRITE(6,636)
  636 FORMAT(1H1,20X,'DOUBLE FOURIER SERIES',// 28X,'RESULT',//,
     1     9X,'I',4X,'Z(I,1)',4X,'Z(I,2)',5X,8X,'X(I)',8X,'QC(I)',/)
        DO 1 I=1,NP
        PX=Z(I,1)
        PY=Z(I,2)
        PAI=0.0
        Q=0.0
        MD=MEN-1
        ND=NEN-1
        MS(1)=MEN
        ME(1)=MEND
        KS(1)=NEN
        KE(1)=NEND
        MS(2)=1
        ME(2)=MD
        KS(2)=NEN
        KE(2)=NEND
        MS(3)=MEN
        ME(3)=MEND
        KS(3)=1
        KE(3)=ND
        MS(4)=40
        ME(4)=40
        KS(4)=NEN
        KE(4)=NEND
        MS(5)=MEN
```

SOURCE STATEMENT

```
      ME(5)=MEND
      KS(5)=50
      KE(5)=50
      MS(6)=1
      ME(6)=MD
      KS(6)=1
      KE(6)=ND
      MS(7)=70
      ME(70)=70
      KS(7)=1
      KE(7)=ND
      MS(8)=1
      ME(8)=MD
      KS(8)=80
      KE(8)=80
      MS(9)=90
      ME(9)=90
      KS(9)=90
      KE(9)=90
      DO 10 L=1,9
      MSL=MS(L)
      MEL=ME(L)
      NSL=KS(L)
      NEL=KE(L)
      MM=0
      DO 2 MF=MSL,MEL
      MM=MM+1
      M=MEL+1-MM
      IF(L.EQ.4.OR.L.EQ.7.OR.L.EQ.9) M=0
      FM=FLOAT(M)
      PML=FM*P/XLE
      PML2=PML*PML
      IF(L.EQ.4.OR.L.EQ.7.OR.L.EQ.9) GO TO 70
      PM2=2.0/(FM*P)
   70 CONTINUE
      IF(JSD.EQ.1) GO TO 800
      CX=DCOS(PML*PX)
      GO TO 801
  800 CX=COS(PML*PX)
  801 CONTINUE
      DO 40 K=1,KCE
      IF(L.EQ.4.OR.L.EQ.7.OR.L.EQ.9) GO TO 400
      IF(JSD.EQ.1) GO TO 802
      SPX(K)=PM2*(DSIN(PML*XE(K))-DSIN(PML*XS(K)))
      GO TO 803
  802 SPX(K)=PM2*(SIN(PML*XE(K))-SIN(PML*XS(K)))
  803 CONTINUE
      GO TO 40
  400 SPX(K)=(XE(K)-XS(K))/XLE
   40 CONTINUE
      NN=0
      DO 3 NF=NSL,NEL
      NN=NN+1
      N=NEL+1-NN
      IF(L.EQ.5.OR.L.EQ.8.OR.L.EQ.9) N=0
      FN=FLOAT(N)
```

SOURCE STATEMENT

```
      PNH=FN*P/YHE
      PNH2=PNH*PNH
      IF(L.EQ.5.OR.L.EQ.8.OR.L.EQ.9) GO TO 71
      PN2=2.0/(FN*P)
   71 CONTINUE
      IF(JSD.EQ.1) GO TO 804
      CY=DCOS(PNH*PY)
      GO TO 805
  804 CY=COS(PNH*PY)
  805 CONTINUE
      QMN=0.0
      DO 41 K=1,KCE
      IF(L.EQ.5.OR.L.EQ.8.OR.L.EQ.9) GO TO 401
      IF(JSD.EQ.1) GO TO 806
      SPY(K)=PN2*(DSIN(PNH*YE(K))-DSIN(PNH*YS(K)))
      GO TO 807
  806 SPY(K)=PN2*(SIN(PNH*YE(K))-SIN(PNH*YS(K)))
  807 CONTINUE
      GO TO 41
  401 SPY(K)=(YE(K)-YS(K))/YHE
   41 QMN=QMN+CQ(K)*SPX(K)*SPY(K)
   12 CXY=CX*CY
      IF(JQ.EQ.0) GO TO 30
      Q=Q+QMN*CXY
   30 IF(JP.EQ.0) GO TO 31
      PMN=QMN/(DX*PML2+DY*PNH2+DK)
      PAI=PAI+PMN*CXY
   31 CONTINUE
    3 CONTINUE
    2 CONTINUE
   10 CONTINUE
      QC(I)=Q
      X(I)=PAI
      WRITE(6,635)(I,Z(I,1),Z(I,2),X(I),QC(I))
  635 FORMAT(1H ,5X,I5,2F10.2,5X,2E12.4)
    1 CONTINUE
CCC   FIGURE START
      IF(JFIG.EQ.0) GO TO 1283
      WRITE(6,650)IMOJI
  650 FORMAT(1H1,5(/),19X,32(3H*  ),//19X,1H*,92X,1H*,//19X,1H*,5X,
     1       20A4,7X,1H*,//19X,1H*,92X,1H*,//19X,32(3H*  ),5(/))
      WRITE(6,6161)DX,DY,DK,MEND,NEND,MEN,NEN
 6161 FORMAT(1H0,30X,'DOUBLE FOURIER SERIES',
     1 //,30X,'DX =',F12.4,5X,'DY =',F12.4,5X,'DK =',F12.6,5X,
     2 //,30X,'MEND =',I5,5X,'NEND =',I5,5X,'MEN =',I5,5X,'NEN =',I5,/)
      WRITE(6,6620)KXE,KYE
 6620 FORMAT(1H0,/,26X,'KXE =',I3,4X,'KYE =',I3,/)
      IF(KXE.GT.18) GO TO 1288
      WRITE(6,6660)
      DO 1981 I=1,KYE
      IF(JFIG)1982,1983,1984
 1982 NS=(I-1)*KXE+1
      NH=I*KXE
      IY=I
       GO TO 1985
 1984 NS=NP-I*KXE+1
```

SOURCE STATEMENT

```
      NH=NP-(I-1)*KXE
      IF(JKE.NE.0) NS=NS-JKE
      IF(JKE.NE.0) NH=NH-JKE
      IY=KYE-I+1
 1985 WRITE(6,6961)IY,(IN,IN=NS,NH)
 6961 FORMAT(1H0,//,I3,1X,21I6)
 1981 CONTINUE
      WRITE(6,6962)(IX,IX=1,KXE)
 6962 FORMAT(1H0,///,4X,21I6)
 1983 CONTINUE
      WRITE(6,6660)
 6660 FORMAT(1H1,//)
      WRITE(6,6963)IMOJI
 6963 FORMAT(1H ,25X,20A4,//)
      DO 1281 I=1,KYE
       IF(JFIG) 1282,1283,1284
 1282 NS=(I-1)*KXE+1
      NE=I*KXE
      IY=I
      GO.TO 1285
 1284 NS=NP-I*KXE+1
      NE=NP-(I-1)*KXE
      IF(JKE.NE.0) NS=NS-JKE
      IF(JKE.NE.0) NE=NE-JKE
      IY=KYE-I+1
 1285 WRITE(6,6661)IY,(X(IN),IN=NS,NE)
 6661 FORMAT(1H0,//,I3,1X,21F6.1)
 1281 CONTINUE
      WRITE(6,6962)(IX,IX=1,KXE)
      GO TO 1283
 1288 WRITE(6,6662)
 6662 FORMAT(1H0,//,20X,'KXE.GT.18 DE FIGURE NASI',/)
 1283 CONTINUE
CCC   FIGURE END
      IF(IPR.EQ.0) GO TO 130
  130 CONTINUE
      RETURN
      END
```

SOURCE STATEMENT

```
C     SUBROUTINE
      SUBROUTINE DATA
      IMPLICIT  REAL*8 (A-H,O-Z)
      COMMON/D0R    /IFK,IFJ
      COMMON/D1R    /KINDU,KINDV,KINDDX,KINDDY,KINDK
      COMMON/D2R    /XSTART,YSTART,XEND,YEND,DMX,DMY,MXE,MYE,KXE,KYE
      COMMON/D2C    /Z(500,2),AREA,SAREA,NP
      COMMON/D3R    /U,V,DX,DY,DK
      COMMON/D7R    /XLE,YHE,JXYE
      COMMON/DR8    /JKE
      COMMON/D12    /JFIG
      COMMON/D12R   /IPR
      COMMON/FUKA   /XS(50),YS(50),XE(50),YE(50),CQ(50),KCE
      COMMON/MNEND /MEND,NEND,MEN,NEN
      COMMON/JSIDO /JSD,JQ,JP
      DIMENSION MOJI(5)
      DIMENSION ZX(100),ZY(100)
  100 READ(5,500)ID,ID1,ID2,ID3,ID4,ID5,MOJI
  500 FORMAT(I2,3X,5I5,30X,5A4)
      IF(ID.LT.1.OR.ID.GT.13) GO TO 110
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13),ID
    1 WRITE(6,601)ID,MOJI
  601 FORMAT(1H0,//,2X,' *',I2,' *',68X,5A4)
      KINDU=ID1
      KINDV=ID2
      KINDDX=ID3
      KINDDY=ID4
      KINDK=ID5
      WRITE(6,610)KINDU,KINDV,KINDDX,KINDDY,KINDK
  610 FORMAT(1H0,3X,'KINDU =',I3,5X,'KINDV =',I3,5X,
     1          'KINDDX =',I3,5X,'KINDDY =',I3,5X,'KINDK =',I3,/)
      GO TO 100
    2 WRITE(6,601)ID,MOJI
      READ(5,502)KXE,KYE,XSTART,YSTART,XEND,YEND
  502 FORMAT(2I5,4F10.0)
      WRITE(6,615 )KXE,KYE,XSTART,YSTART,XEND,YEND
  615 FORMAT(1H0,3X,5HKXE =,I4,5X,5HKYE =,I4,5X,5HXSTART =,F10.4,5X,
     1       8HYSTART =,F10.4,5X,6HXEND =,F10.4,5X,6HYEND =,F10.4,/)
      MXE=KXE-1
      MYE=KYE-1
      IF(KXE.EQ.1) GO TO 900
      DMX=(XEND-XSTART)/FLOAT(MXE)
      GO TO 901
  900 DMX=0.0
  901 CONTINUE
      IF(KYE.EQ.1) GO TO 902
      DMY=(YEND-YSTART)/FLOAT(MYE)
      GO TO 903
  902 DMY=0.0
  903 CONTINUE
      NP=KXE*KYE
      NE4=MXE*MYE
      SAREA=DMX*DMY
      AREA=SAREA*FLOAT(NE4)
      WRITE(6,620)DMX,DMY,NP,NE4,AREA,SAREA
  620 FORMAT(1H0,/,5X,'DMX =',F12.4,5X,'DMY =',F12.4,5X,
```

SOURCE STATEMENT

```
      1      'NP =',I5,5X,'NE4 =',I5,5X,'AREA =',F12.5,5X,'SAREA =',F12.4,
     2      /)
      ZY(1)=YSTART
      ZX(1)=XSTART
      IF(KYE.EQ.1) GO TO 904
      DO 23 I=2,KYE
      I1=I-1
   23 ZY(I)=ZY(I1)+DMY
  904 CONTINUE
      IF(KXE.EQ.1) GO TO 200
      DO 22 J=2,KXE
      J1=J-1
   22 ZX(J)=ZX(J1)+DMX
  200 DO 24 I=1,KYE
      DO 24 J=1,KXE
      K=(I-1)*KXE+J
  202 Z(K,1)=ZX(J)
   24 Z(K,2)=ZY(I)
      GO TO 100
    3 WRITE(6,601)ID,MOJI
      READ(5,530)U,V,DX,DY,DK
  530 FORMAT(5F10.0)
      WRITE(6,630)U,V,DX,DY,DK
  630 FORMAT(1H0,3X,'U =',F10.2,5X,'V =',F10.2,5X,
     1      'DX =',F12.4,5X,'DY =',F12.4,5X,'DK =',F12.5 ,/)
      GO TO 100
    4 WRITE(6,601)ID,MOJI
      KCE=ID1
      KDE=ID2
      WRITE(6,640)KCE,KDE
  640 FORMAT(1H0,3X,'KCE =',I3,5X,'KDE =',I5,/)
      READ(5,540)(XS(I),YS(I),XE(I),YE(I),CQ(I),I=1,KCE)
  540 FORMAT(5F10.0)
      WRITE(6,642)(I,XS(I),YS(I),XE(I),YE(I),CQ(I),I=1,KCE)
  642 FORMAT(1H ,/(5X,I5,5F16.7))
      GO TO 100
    5 WRITE(6,601)ID,MOJI
      MEND=ID1
      NEND=ID2
      MEN=ID3
      NEN=ID4
      WRITE(6,650)MEND,NEND,MEN,NEN
  650 FORMAT(1H0,3X,'MEND =',I5,5X,'NEND =',I5,5X,'MEN =',I5,5X,'NEN =',
     1 I5,/)
      GO TO 100
    6 WRITE(6,601)ID,MOJI
      JSD=ID1
      JP=ID2
      JQ=ID3
      WRITE(6,660)JSD,JP,JQ
  660 FORMAT(1H0,3X,'JSD =',I5,5X,'JP =',I5,5X,'JQ =',I5,5X,/)
      GO TO 100
    7 WRITE(6,601)ID, MOJI
      JXYE=ID1
      WRITE(6,670)JXYE
  670 FORMAT(1H0,3X,'JXYE =',I3,/)
```

SOURCE STATEMENT

```
      IF(JXYE.EQ.0) GO TO 100
      READ(5,570)XLE,YHE
  570 FORMAT(2F10.0)
      WRITE(6,671)XLE,YHE
  671 FORMAT(1H0,3X,'XLE =',F10.2,5X,'YHE =',F10.2,/)
      GO TO 100
    8 WRITE(6,601)ID,MOJI
      JKE=ID1
      WRITE(6,680)JKE
  680 FORMAT(1H0,3X,'JKE =',I5,/)
      IF(JKE.EQ.0) GO TO 100
      NPJ=NP+1
      NP=NP+JKE
      READ(5,581)(Z(I,1),Z(I,2),I=NPJ,NP)
  581 FORMAT(8F10.0)
      WRITE(6,681)(I,Z(I,1),Z(I,2),I=NPJ,NP)
  681 FORMAT(1H ,/(4(5X,I5,2F10.2)))
      GO TO 100
    9 GO TO 100
   10 GO TO 100
   11 GO TO 100
   12 WRITE(6,601)ID,MOJI
      IPR=ID1
      JFIG=ID2
      WRITE(6,6612)IPR,JFIG
 6612 FORMAT(1H0,3X,'IPR =',I3,5X,'JFIG =',I3,/)
      IF(IPR.EQ.0) GO TO 100
      GO TO 100
   13 WRITE(6,6199)ID
 6199 FORMAT(1H0,//,2X,2H* ,I2,2H *,5X,11HDATA FINISH,/)
      GO TO 1001
  110 WRITE(6,6110)ID
 6110 FORMAT(1H0,//,2X,2H* ,I2,2H *,5X,9HID OKASII,/)
 1001 CONTINUE
      RETURN
      END
```

VITA

The author was born in Yamaguchi prefecture, on March 2, 1944. He was brought up in a seaside village in Ehime prefecture. He received the Bachelor of Engineering degree in civil engineering in 1967 and the Master of Engineering degree in civil engineering in 1969 from Kobe University. From 1969 to 1972, he was employed as a civil engineer with Yachiyo Engineering Co. Ltd., Tokyo.

He began his doctorate course in the Graduate School at Kyoto University in 1972. His major was environmental planning, analysis and management; his minor was systems engineering and structural mechanics.

In July-August, 1970, the author attended the 5th International Water Pollution Research Conference held in San Francisco to deliver the paper "Dynamic Programming for a Sewage Treatment System". In July-August, 1975, he attended the 16th Congress of the International Association for Hydraulic Research held in São Paulo to deliver the paper "Finite Element & Linear Programming Method and Water Pollution Control".

In January, 1976, the author was invited to participate in the activities of the Task Committee on Systems Analysis Techniques which belongs to the Committee on Water Resources Systems of the International Association for Hydraulic Research.

He is a member of the Japan Society of Civil Engineers and the International Association for Hydraulic Research.